Universidad
Carlos III de Madrid
www.uc3m.es

Escuela Politécnica Superior - Grado en Ingeniería Telemática

EVALUACIÓN DE MECANISMOS DE SOPORTE DE TRÁFICO MULTICAST CON MOVILIDAD BASADA EN RED

Trabajo Fin de Grado

Autor: Sergio González Díaz

Tutor: Carlos Jesús Bernardos Cano

Directores: Pablo Serrano Yáñez-Mingot

Luis Miguel Contreras Murillo

Junio de 2015

Agradecimientos

Primeramente quería agradecer a mis padres, mi hermana, mis abuelos, mi tía y mi novia el apoyo mostrado cada día y su forma de ser, nada de esto habría sido posible sin ellos. En especial quería mencionar a mis padres por brindarme la oportunidad de estudiar una carrera.

Dar las gracias también a todos mis compañeros y en especial a Carlos, Mario, José Manuel, Alberto y Pablo por estar a mi lado a lo largo de todo esta etapa académica de mi vida, sin ellos todo habría sido mucho más complicado.

Por último, agradecer a mi tutor Carlos Jesús y a mi director Luis Miguel toda la dedicación y ayuda en la realización de este trabajo.

<< Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber. >>

Albert Einstein (1879 - 1955)

<< No he fracasado. He encontrado 10.000 soluciones que no funcionan. >>

Thomas Alva Edison (1847 – 1931)

Resumen

En la década de los 70, durante el comienzo del desarrollo del modelo TCP/IP, todas las conexiones de datos se realizaban a través de redes cableadas. En el pasado año 2014, según un estudio de CISCO [1], el tráfico mundial de datos móviles creció un 69%, consumiendo un total de 30 Exabytes. El mismo estudio muestra que el tráfico global de datos móviles se multiplicará por diez en los próximos cinco años llegando a los 292 Exabytes anuales en 2019.

Estos datos muestran que el uso de Internet en los dispositivos móviles está aumentando a pasos agigantados. La gran ventaja que ofrecen este tipo de dispositivos es la posibilidad de conectarse desde cualquier lugar, moverse de un sitio a otro y mantener la conexión. Esto a su vez es el mayor problema de las conexiones inalámbricas, ya que muchas veces el desplazamiento provoca un cambio de punto de acceso, después del cual se debe mantener la conexión.

Para solventar este problema se comenzó el desarrollo de protocolos de movilidad como MIPv6 [2] (Mobile IPv6). Este protocolo permite dotar de movilidad a un nodo dentro de una red IPv6. Cada nodo móvil dentro de la red tiene siempre asociada una dirección permanente llamada HoA (*Home Address*), independientemente de su punto de acceso. Cuando este nodo móvil se sitúa lejos de su red hogar se le asigna una dirección temporal llamada CoA (*Care-of Address*) que ofrece información a cerca de la localización del nodo.

Para gestionar la movilidad de los nodos y permitir su comunicación fuera de su red hogar surge una entidad, el *Home Agent* (HA). El HA es el router en el hogar, donde el nodo móvil ha registrado su CoA, este interceptará todos los paquetes destinados a su HoA para encapsularlos en un túnel hacia su CoA. Gracias a estas dos direcciones y esta entidad el nodo puede moverse de su red origen pasando de una red a otra y manteniendo su conexión y resolviendo el dilema de la movilidad en red.

Para mejorar la gestión que propone MIPv6 se han desarrollado diferentes protocolos cuya ventaja es permitir movilidad IP dentro de la red, pero en este caso, sin ningún tipo de requerimiento en la señalización de movilidad por parte del nodo móvil. La propia red gestiona la movilidad del nodo, lo localiza en la red e inicia la señalización. De esta manera la movilidad se realiza facilitando las labores al nodo móvil. Con estas premisas nació PMIPv6 [3] (Proxy Mobile IPv6), basado en MIPv6, el único protocolo de movilidad basado en red estandarizado por la IETF. El gran beneficio de los protocolos basados en red es que la propia red es la que gestiona la movilidad, y en el caso de PMIPv6, esta movilidad

es transparente al nodo móvil ya que siempre mantiene la misma dirección IPv6, a diferencia de MIPv6 donde la CoA cambia en cada red foránea.

Para la implementación de PMIPv6 en el nodo móvil no hace falta ningún tipo de software adicional ya que para él todo es totalmente transparente, le será asignada una dirección IPv6 que mantendrá durante toda la conexión, aunque cambie el punto de acceso. Todo esto se produce gracias a los agentes que controlan la movilidad en la red que son el LMA (*Local Mobility Anchor*) y el MAG (*Mobile Access Gateway*).

El LMA es el equivalente al HA en MIPv6, es la entidad encargada de manejar toda la movilidad de la red. El MAG hace la función de router de acceso y es el encargado de localizar el nodo móvil y señalizar sus movimientos al LMA. El MAG y el LMA se comunican de manera continua a través de un túnel.

El funcionamiento de PMIPv6 es muy sencillo, cuando un nodo móvil entra en un dominio IPv6 y se conecta al enlace, el MAG comprueba si está autorizado para el uso del servicio de movilidad basado en red, si lo está entonces le asigna una dirección IPv6 basada en el HNP (*Home Network Prefix*), un prefijo de su red hogar. Esta dirección IPv6 se mantendrá aunque el nodo móvil se mueva a otro MAG dentro de dominio PMIPv6. El LMA también se encarga de de guardar información de la conexión como: un identificador del nodo móvil, una dirección del MAG, un identificador del tunel entre el LMA y el MAG entre otras cosas.

Con el auge actual de Internet y el incremento en el uso de dispositivos móviles ha aumentado el consumo de contenido multimedia, del cual cabe destacar el streaming de vídeo, el vídeo bajo demanda y la IPTV. Muchos de estos servicios hacen uso de Multicast, un método de transmisión de datos a múltiples redes y destinatarios de manera simultánea. Para poder realizar transmisiones de datos multicast hay direcciones IP reservadas, tanto en IPv4 como en IPv6. El funcionamiento de multicast es muy sencillo, el host que quiere recibir tráfico manda un mensaje a su router y se suscribe al grupo multicast (IP dentro de un rango reservado), a partir de ahí el router ya sabe que tiene que enviar el tráfico multicast de ese grupo a ese host en particular. El protocolo multicast en el que nos vamos a centrar es MLD (*Multicast Listener Discovery*). MLD es una herramienta que utilizan los routers IPv6 para descubrir subscriptores multicast en un enlace directo, es el equivalente a IGMP en IPv4. Se usará la versión más reciente de este protocolo MLDv2 [4].

El objetivo final de este Trabajo Fin de Grado es evaluar estos mecanismos de envío de tráfico multicast (MLD), en un escenario real bajo un protocolo de movilidad basado en red (PMIPv6). Para desplegar este escenario se utilizará un prototipo con routers empleando la distribución OpenWRT [5].

OpenWRT es una distribución de Linux basada en firmware destinada al uso en routers. Esta distribución permite en muchos casos aprovechar más a fondo las características y especificaciones de nuestros routers en comparación al firmware instalado por el fabricante. OpenWRT inicialmente ofreció soporte al modelo Linksys WRT54G únicamente, debido a su auge y popularidad en un periodo corto de tiempo esta distribución se expandió a otros dispositivos de otras marcas como Asus, Netgear, D-Link entre otras.

OpenWRT es un proyecto de software libre y, gracias a ello, hay gran cantidad de información y soporte a cerca de esta distribución en comunidades y foros de Internet. Está disponible la web oficial openwrt.org donde se puede consultar documentación, tutoriales, dispositivos soportados y proceder a la descarga de la distribución.

La gran ventaja que ofrece OpenWRT es un sistema de archivos totalmente modificable con gestión de paquetes, esto permite la personalización de la distribución del router incluyendo sólo los paquetes deseados, añadiendo o eliminando funciones para un mejor rendimiento. Para los desarrolladores OpenWRT proporciona muchas ventajas ya que pueden crear una aplicación sin tener que desarrollar el firmware completo para el router. Para la gestión y manejo de OpenWRT se utiliza principalmente la linea de comandos bajo una conexión remota como puede ser telnet aunque también dispone de interfaz web.

Debido al auge y crecimiento de OpenWRT ha surgido una variante llamada dd-wrt.

Para llegar al objetivo del Trabajo Fin de Grado se comenzó con una familiarización con el entorno de OpenWRT en diferentes modelos de routers, inicialmente con Linksys WRT54GL [6]. De manera preliminar se procedió a instalar una imagen genérica de OpenWRT descargada de la web. La distribución elegida fue la última versión de OpenWRT, Barrier Breaker [7], debido a que es la versión más reciente y ofrece más ventajas y compatibilidades que versiones anteriores, esta imagen básica iba a permitir ver el funcionamiento de este tipo distribución Linux en el dispositivo para poder ver el manejo y facilidades que ofrece.

Después de la familiarización con el entorno se procedió a crear una imagen a medida para nuestro router, eliminando funciones inservibles para nuestro uso como puede ser el firewall y añadiendo funcionalidades esenciales para el propósito final como es el soporte total IPv6 y 802.11 para tener conexión inalámbrica al router desde nuestro nodo móvil. Debido al añadido de paquetes la imagen se volvió muy pasada para el hardware del Linksys WRT45GL por lo que optamos por usar una versión anterior a Barrier Breaker, más ligera para que pudiera usarse de manera ágil en el router. La versión escogida fue la anterior, Attitude Adjustment. Se creó la imagen de manera exitosa por lo que se procedió a

realizar la compilación cruzada [8] del código de PMIPv6 para hacer funcionar un escenario básico formado por:

- Un LMA

- Dos MAG

- Un PC conectado por cable al LMA

- Un PC conectado inalámbricamente a un MAG (nodo móvil)

Una vez se comprobó el correcto funcionamiento del escenario propuesto se procedió a dar el siguiente paso, hacer funcionar un proxy multicast para que el tráfico multicast pudiese atravesar el escenario. El programa elegido fue mcproxy [9] con soporte para OpenWRT.

Surgieron algunas dificultades a la hora de hacer funcionar PMIPv6 y mcproxy simultáneamente en los Linksys WRT45GL debido a su hardware, por lo que finalmente se utilizó el modelo Asus WL-500G Premium v1 [10] con tarjeta de red Atheros. Este modelo con un hardware más potente que el Linksys WRT45GL ya permitía el uso simultáneo de los dos programas de manera ágil, por lo que se procedió al montaje del escenario final, exactamente igual al descrito anteriormente donde el PC conectado por cable sería el emisor de tráfico multicast y el nodo móvil el receptor.

Una vez que el escenario funcionó correctamente se comenzaron a hacer pruebas para evaluar el cambio de MAG con un envío continuo de tráfico multicast, estas pruebas sirven para demostrar la mala implementación del cambio. Para ver la eficiencia se probó la continuidad del tráfico multicast con diferentes programas. Para el envío y recepción de tráfico multicast de manera sencilla y para comprobar el correcto funcionamiento del escenario se utilizó mcsender en el emisor y mcfirst en el receptor. Estos dos programas tienen un funcionamiento similar al ping, realizan un envío de tráfico de manera continua con una marca de tiempo en cada paquete recibido. Esto nos ayuda a medir el tiempo que pasa el nodo móvil sin recibir tráfico multicast en un cambio de MAG.

En MLD existen diferentes temporizadores para gestionar la conexión, estos tienen un valor muy elevado lo que implica que durante el cambio de MAG se produce una pérdida de paquetes, esto se produce por el largo periodo de tiempo correspondiente al vencimiento de los temporizadores.

Para solucionar este problema se decidió implementar la RFC 6636 [11]. Este documento propone un pequeño cambio en los temporizadores para optimizar la recepción de tráfico multicast en un cambio de MAG. Este cambio de los temporizadores hace que el

tráfico multicast se redirija hacia la situación actual del nodo móvil de manera más rápida. Estos temporizadores se modificaron en el código fuente del programa mcproxy de manera acorde a la definición marcada en la RFC 6636 [11]. Una vez fueron ajustados se realizaron una serie de pruebas, con las herramientas antes mencionadas, para poder medir el impacto de la variación de los temporizadores en el envío de tráfico multicast en nuestro escenario.

Para finalizar esta memoria, se muestran diferentes gráficas que evidencian la mejora que ha producido la implantación de la RFC 6636 [11] respecto a los temporizadores originales en el cambio de MAG con envío de tráfico multicast.

Abstract

In the 70's, during the development of the TCP/IP model, all the data connections were wired. In the past year 2014, a CISCO [1] investigation reveals that the global mobile data traffic grew 69%, consuming a global amount of 30 Exabytes. According to this research, the global mobile data traffic will increase by ten times in the next five years, reaching 292 Exabytes per year in 2019.

This data shows the increasing use of Internet on smartphones. The great advantage offered by these devices is the possibility to have an Internet connection anywhere, move and maintain the connection. This advantage is also the main problem in the wireless connections, because sometimes the movement causes a change of access point, and then the connection must be maintained.

To solve these problems triggered the development of mobility protocols like MIPv6 [2] (Mobile IPv6). This protocol can provide mobility to a mobile node into an IPv6 network. Each mobile node into the network has a permanent address called HoA (Home Address). When the mobile node moves far of his home network a temporary address called CoA (Care-Of Address) is assigned, this address offers information about the mobile node's location.

To manage the node's mobility and allow the communication outside your home network, the role of the Home Agent (HA) is introduced. The HA is the router in the home network, where the mobile node register its CoA, this entity will intercept all the data packages in destination to the mobile node's HoA to resend it in a bidirectional tunnel in direction to its CoA. Thanks to these two addresses and this agent the mobile node can move out from its home network to a foreign network maintaining the connection and resolving the network mobility dilemma.

To improve the management proposed by MIPv6, different protocols have been developed whose advantage is to allow IP mobility in the network, but in this case, without any mobility signalling requirement on the mobile node. The network by itself manages the mobility, tracking and signalling of the mobile node. It is a way to make mobility easier for the mobile node. With this premises PMIPv6 [3] (Proxy Mobile IPv6) was born, based on MIPv6, and it is the only network-based mobility management protocol standardized by the IETF. The great benefit of network-based protocols is that the network manage mobility itself, and in the case of PMIPv6 the mobility is transparent for the mobile node, because it maintains the same IPv6 address, unlike MIPv6 where the CoA changes on each foreign network.

To implement PMIPv6 in the mobile node there is no need of any additional software, it is completely transparent for the mobile node. The MN is assigned an IPv6 address that will be maintained during the connection, although it change the access point. All of this is possible thanks to two agents that control the network mobility, LMA (Local Mobility Anchor) and MAG (Mobility Access Gateway).

LMA is equivalent to HA in MIPv6, it is the entity responsible of managing the entire network mobility. The MAG acts as an access router and is responsible to locate the mobile node and signal its movements to LMA. MAG and LMA communicates between them continuously through a tunnel.

The operation of PMIPv6 is simple, when a mobile node enters in a PMIPv6 domain and connects to the network, the MAG checks if it is authorised to use the network-based mobility service, if it is, an IPv6 address based in the HNP (Home Network Prefix) is assigned. This IPv6 address will be maintained even if the mobile node moves to other MAG into the PMIPv6 domain. The LMA has the function of save information of the connection like: mobile node's identifier, MAG address, identifier of the tunnel between the LMA and the MAG among other things.

With the current boom of Internet and the rising use of mobile devices, the media content consumption has increased, including video streaming, video on demand and IPTV. Many of this services use Multicast. Multicast is a method of sending IP datagrams to a group of interested receivers in a single transmission. To do multicast data transmissions are reserved a range of IP addresses, both in IPv4 and in IPv6. Multicast operation is simple, the host who wants to receive traffic sends a message to its router and it subscribes to the multicast group (IP's reserved range). Then the router knows how to send multicast traffic for that group to that particular host. We are going to focus in the multicast protocol MLD (Multicast Listener Discovery). MLD is tool used by IPv6 routers to discover multicast listeners in a direct link, it is equivalent to IGMP in IPv4. We focus on the latest version of this protocol, MLDv2 [4].

The final objective of this Bachelor Thesis is to evaluate this multicast sending mechanisms (MLD), in a real scenario running a network-based mobility management protocol (PMIPv6). To deploy the scenario we will utilize a prototype with routers using a OpenWRT[5] distribution.

OpenWRT is a Linux distribution for use in routers. This distribution allows to take advantage of the features and specification of our routers compared with the original factory firmware. Initially OpenWRT offered support for one single model, Linksys WRT54G, because

of its popularity in a short period, it expanded to other brands and devices like Asus, Netgear or D-Link.

OpenWRT is an Open Source software, and thanks to that, it has a lot of documentation and support in Internet forums and communities. There is an official web: openwrt.org where you can consult information, documentations, tutorials, supported devices and download the distribution.

The great advantage of OpenWRT is that it provides a fully writable filesystem with package management, this allow to customize the router's distribution including only the desired packages, adding or removing features for a better performance. For the developers OpenWRT provides a lot of advantages because they can build an application without having to build a complete firmware for the router. For OpenWRT management the user can use a virtual terminal connection like telnet or ssh although there is a web interface too.

Thanks to the boom of OpenWRT a new distribution variant has emerged called dd-wrt.

To achieve the objective of this Bachelor Thesis we started with the familiarization with the OpenWRT environment with the router model Linksys WRT54GL [6]. Preliminarily I proceeded to install a generic OpenWRT image downloaded from the official web. The chosen distribution was Barrier Breaker [7], because it is the latest version and offers more compatibility and advantages than the previous versions. This basic image would allow to see the performance of this Linux distribution in the Linksys WRT45GL to see the facilities offered.

After getting familiar with the enviroment I proceeded to create a custom router image, removing useless functions like firewall and adding essential functions for the final purpose like IPv6 total support and 802.11 for the mobile node's wireless connection. Due to this adding, the image became heavier for the Linksys WRT45GL hardware and it produced some freezes and disconnections. A previous version of OpenWRT was chosen to improve the performance, Attitude Adjustment. The performance with this version was satisfactory so I proceeded with the PMIPv6 code crosscompilation [8] to run a basic scenario with:

- One LMA
- Two MAG
- One computer with wired connection with LMA
- One computer with wireless connection with MAG

Once the correct working in this scenario was checked the next step began, which consisted in running a proxy multicast in addition to PMIPv6. This will provide the transmission of the multicast traffic along the scenario, allowing the communication between the two computers across the network. The chosen program was mcproxy [9], because of the OpenWRT support.

Along the work different problems arose when running simultaneously both programs because of the Linksys WRT45GL limited hardware. Finally the router model chosen was Asus WL-500G Premium v1 [10] with Atheros wireless card. This model is more powerful than the Linksys WRT45GL, and because of that, it allows to run smoother the two programs simultaneously. So I proceeded to mount the final scenario with the Asus router, where the wired computer was the multicast traffic sender and the wireless computer the receiver.

Once the scenario worked properly the evaluation phase started. The main test was changing the MAG while multicast traffic is sent, this test proved the bad implementation of the change of access point (MAG). The programs used for the multicast traffic send and reception were mcsender (sender) and mcfirst (receiver). Both programs have a similar function to ping, they produce a continuous multicast traffic with a timestamp in the reception of each packet. It is very helpful to measure the amount of time that the mobile node does not receive multicast traffic during the handoff.

In MLD there are different timers to manage the connection, this timers have a hight value, this implies that during the MAG handoff a packet loss occurs, this is produced because of the expiration of these timers.

To solve this problem RFC 6636 [11] was implemented. This document proposes a change in the timers to optimize the reception of multicast traffic in the MAG handoff. This change makes a faster redirection of the multicast traffic. These timers were modified in the mcproxy source code according to the RFC 6636 definition. After the timers were adjusted like in the RFC 6636 [11] I did different tests, with the different tools aforementioned, to measure the timer variation impact in the multicast traffic in our scenario.

To finish this Bachelor Thesis, we sho different graphics to demonstrate the improvement of the RFC 6636 [11] implementation in comparison to the original timers.

Índice General

Índice de Figuras

Índice de Tablas

1. Introducción

1.1. Introducción

Este documento presenta una evaluación de los mecanismos de soporte de tráfico multicast con movilidad basada en red. En este capítulo se detalla la motivación que nos ha llevado a plantear y desarrollar este Trabajo Fin de Grado, así como los objetivos del mismo y la estructura del documento.

1.2. Motivación

Actualmente el uso de Internet está creciendo a pasos agigantados, lo que poco a poco se ha traducido en el agotamiento del número de direcciones IPv4 disponibles, todo debido a la mala gestión y asignación de estas unido al escaso espacio de direccionamiento. Actualmente las direcciones IPv4 se encuentran agotadas [12], y aunque existen diferentes mecanismos y herramientas para solventar este problema, como la configuración dinámica de direcciones (DHCP [13]) o el uso de una dirección pública para varios equipos (NAT [14]), este hecho ha producido un interés en IPv6, que permaneció un poco abandonada desde su publicación en 1998.

IPv6 [15] introduce muchas novedades respecto a IPv4 [16], la más destacada quizás sea el aumento del espacio de direccionamiento de 32 a 128 bits, pero hay dos mejoras que resultan muy relevantes en la realización de este Trabajo Fin de Grado:

- Soporte obligatorio de tráfico multicast.
- Proporciona facilidades para el soporte de movilidad IP.

El protocolo IPv6 [17], tiene definido en la especificación base el soporte obligatorio de multicast, a diferencia de IPv4 [16] donde es opcional. IPv6 también tiene un nuevo sistema de autoconfiguración de direcciones [18], que es de gran ayuda en la movilidad a nivel IP.

Hace algo más de una década la gran mayoría de los dispositivos de conexión a Internet eran estáticos, conectados a través de un cable. Hoy en día, la mayoría de dispositivos funcionan de manera inalámbrica, ya sea por WiFi o Internet móvil (GPRS, UMTS, LTE). La utilización de sistemas móviles como portátiles, tabletas o smartphones han producido un rápido crecimiento en las comunicaciones móviles.

Gracias a diferentes estudios realizados por la CNMC [19] [20] [21] (Comisión Nacional del Mercado de las Comunicaciones) podemos ver la evolución del mapa de cobertura móvil 3G y 4G en España:

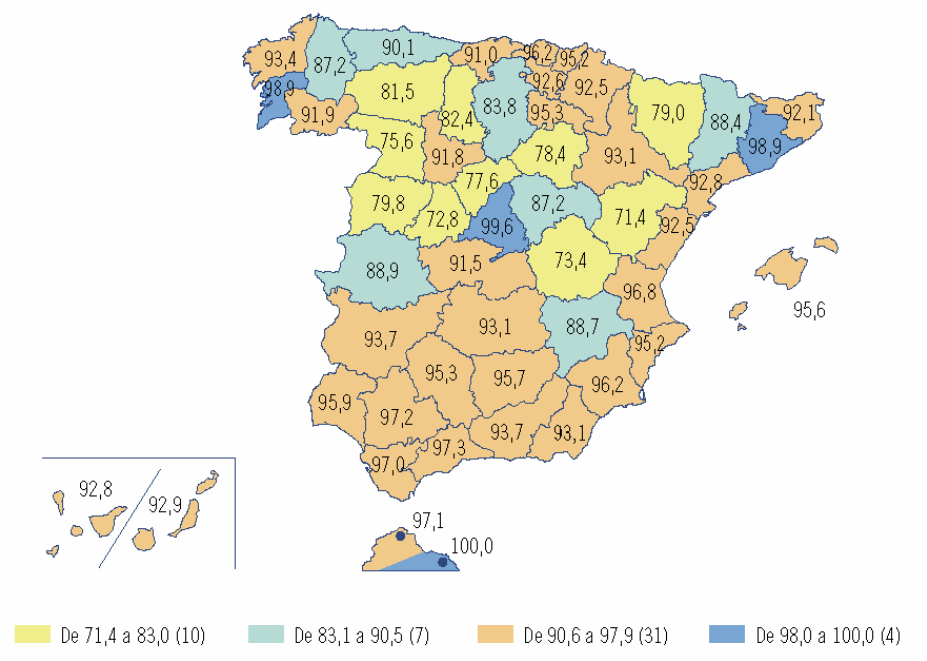


Ilustración 1: Mapa cobertura 3G España en 2011 [19]

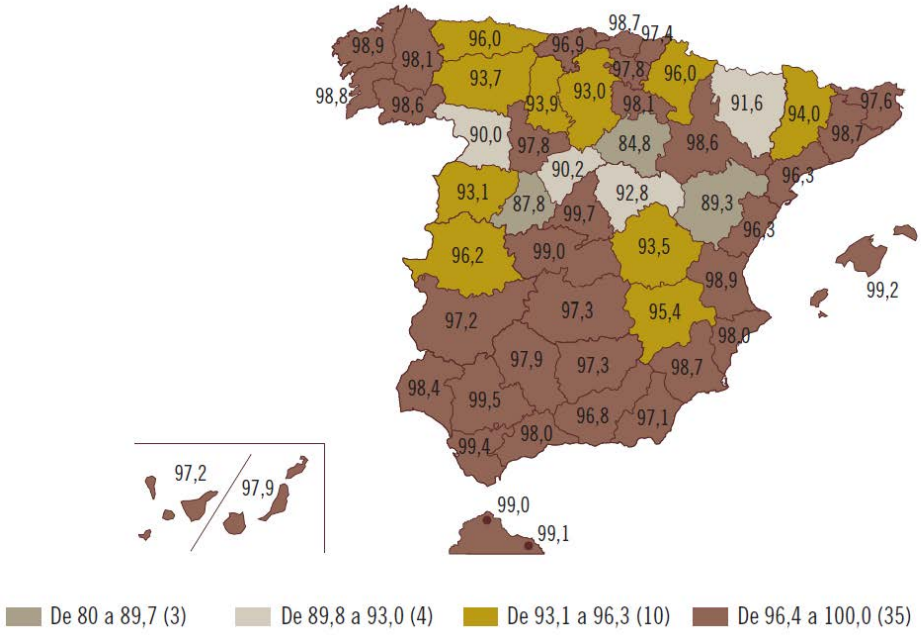


Ilustración 2: Mapa de cobertura 3G España en 2013 [20]

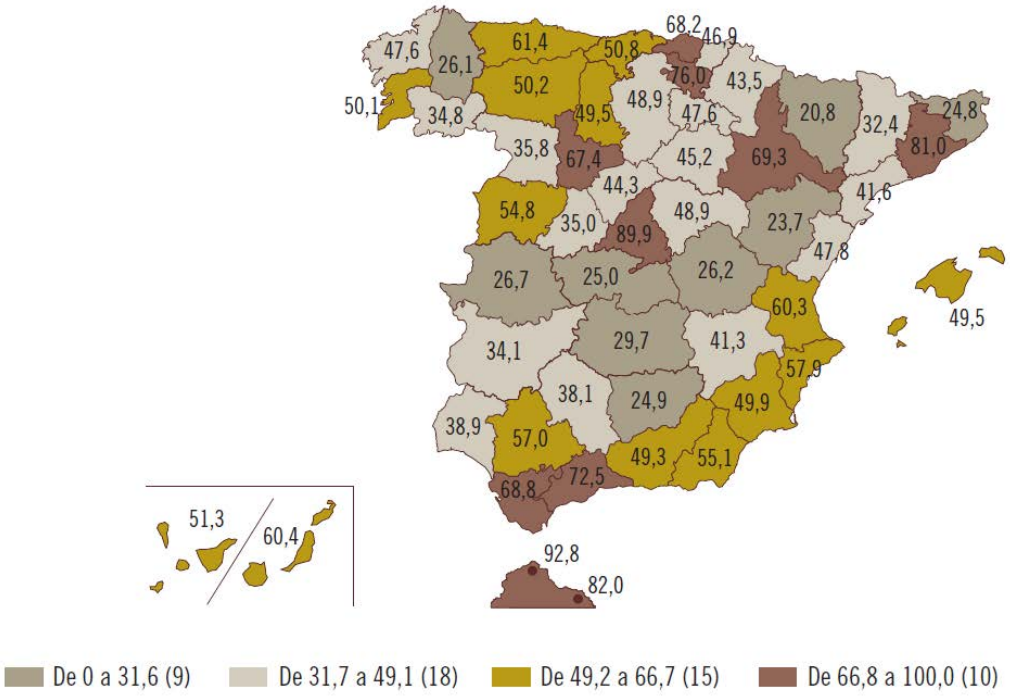


Ilustración 3: Mapa de cobertura 4G España en 2013 [20]

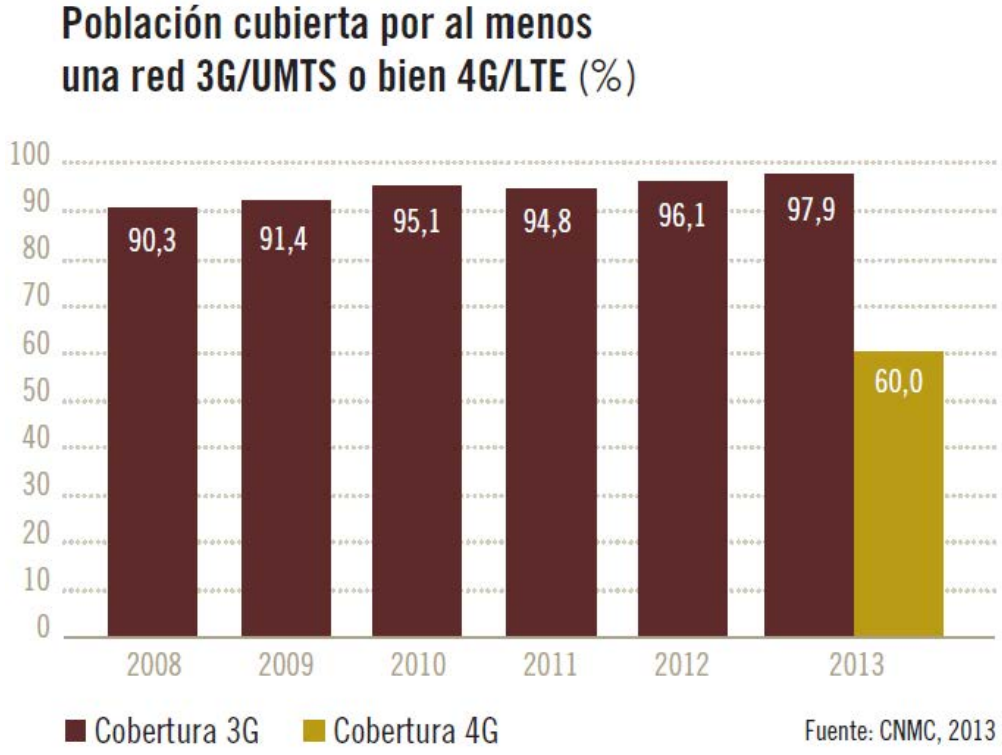


Ilustración 4: Evolución de la cobertura 3G y 4G en España [21]

Como muestran los gráficos el crecimiento de las redes móviles UMTS y LTE es muy alto, llegando actualmente a tener casi un 100% de cobertura en muchos sitios de España.

El gran auge de los smartphones ha propiciado este crecimiento, teniendo la gran ventaja de mantener la conexión mientras nos desplazamos. Todo esto es posible gracias a diferentes protocolos de movilidad que gestionan el movimiento de un punto de acceso a otro manteniendo nuestra conexión. Dentro de estos protocolos de movilidad cabe destacar los protocolos de movilidad basados en red, que gestionan toda esa movilidad de manera transparente al host que se desplaza. Como ejemplo de este tipo de protocolos tenemos PMIPv6, en el que está centrado este Trabajo Fin de Grado y GTP, conceptualmente muy similar a PMIPv6 en la gestión de la movilidad. GTP es el más utilizado actualmente.

Gran parte del tráfico consumido en las redes móviles es video, según un estudio realizado por CISCO [1] en el año 2014 alcanzó el 55% del tráfico móvil mundial y sus previsiones muestran un 72% en 2019. Este consumo de video se produce por la popularidad de plataformas de emisión como YouTube, redes sociales, plataformas de video bajo demanda (VoD) e IPTV.

Actualmente en España operadores de telecomunicaciones como Movistar tienen sistema de televisión IP [22] y esta funciona mediante multicast, un método de transmisión de datos a múltiples redes y destinatarios de manera simultánea.

Por lo tanto cabe destacar la transmisión de tráfico multicast combinado con el uso de un protocolo de movilidad basado en red como PMIPv6 y su utilidad actualmente en redes UMTS/LTE. La unión de todos estos factores, su continuo crecimiento y la posibilidad de análisis y mejora de estos ha motivado la realización de este Trabajo Fin de Grado.

1.3. Objetivos

El principal objetivo de este Trabajo Fin de Grado es poder evaluar la transmisión y recepción de tráfico multicast bajo un protocolo de movilidad basado en red para posteriormente mejorar el funcionamiento. Para alcanzar dicho objetivo se han realizado distintas actividades que podemos dividir en diferentes fases:

- Estudio de IPv6 [17], PMIPv6 [3] y multicast (MLDv2 [4]) para la realización del proyecto.
- Familiarización con el entorno OpenWRT [5], creación de una imagen a medida con el buildroot [23] [24] para los routers y comprobación del correcto funcionamiento de los routers con esta distribución.
- Funcionamiento de un protocolo de movilidad basado en red en los routers. Para ello se tiene que compilar de forma cruzada [8] el código de PMIPv6 para la distribución OpenWRT. Se creará un escenario básico formado por tres routers y dos ordenadores donde funcionará correctamente el protocolo.
- Adición de funcionalidades multicast IPv6 a los routers con OpenWRT. Para ello utilizaremos el programa mcproxy [9] y se adecuará para el correcto funcionamiento en los routers.
- Unión de las dos fases anteriores haciendo funcionar simultáneamente en los routers PMIPv6 y mcproxy.
- Evaluación de la transmisión de tráfico multicast mientras se realiza un cambio de punto de acceso.
- Implementación de la RFC 6636 [11] para mejorar el cambio de punto de acceso en nuestro escenario.
- Evaluación cualitativa y cuantitativa y comparación de las mejoras derivadas de la implementación de la RFC 6636 [11].

1.4. Estructura de la memoria

En esa sección se detalla la estructuración del contenido de este documento para facilitar su comprensión. Este documento se divide en 8 capítulos:

Capítulo 1. Introducción: en este capítulo se realiza una introducción al Trabajo expuesto en este documento detallando las motivaciones que han propiciado su desarrollo junto a los objetivos marcados.

Capítulo 2. Introduction: en este capítulo se realiza en lengua inglesa una introducción al Trabajo expuesto en este documento detallando las motivaciones que han propiciado su desarrollo junto a los objetivos marcados.

Capítulo 3. Estado del Arte: en este capítulo se detallan los protocolos destacados en los que se basa este trabajo.

Capítulo 4. Trabajo Realizado: en este capítulo se explica de manera detallada el núcleo del trabajo realizado en este proyecto.

Capítulo 5. Resultados: en este capítulo se muestran los resultados obtenidos en la realización de este Trabajo Fin de Grado

Capítulo 6. Conclusiones y Futuros Trabajos: en este capítulo se exponen las reflexiones y conclusiones obtenidas así como futuras mejoras aplicables a este Trabajo Fin de Grado.

Capítulo 7. Conclusions and Future Works: en este capítulo se exponen en lengua inglesa las reflexiones y conclusiones obtenidas así como futuras mejoras aplicables a este Trabajo Fin de Grado.

Capítulo 8. Anexos: en este capítulo se explican la planificación, los recursos utilizados y el presupuesto junto a diferentes herramientas utilizadas.

2. Introduction

2.1. Introduction

This document presents an evaluation of the mechanisms of multicast traffic support with network-based mobility. This chapter details the motivation that led us to propose and develop this Final Bachelor Project alongside the objectives and the document structure.

2.2. Motivation

At present the use of Internet is growing up which is translated in the depletion of the number of IPv4 available addresses because of the poor assignation of this limited address space. Nowadays the IPv4 addresses are exhausted [12], and although there are different tools and mechanism to avoid this problem, like dynamic address configuration (DHCP [13]) or the use of one public IP address for different hosts (NAT [14]), this fact has produced a growing interest in IPv6, the last generation of the IP protocol, abandoned since its publication in 1998.

IPv6 [17] Introduces many new features compared to IPv4 [16], perhaps the main feature of IPv6 the increase the address space from 32 to 128 bits, but there are two improvements that are relevant in this Final Bachelor Project:

- Multicast support required.
- Facilities to provide IP mobility support.

IPv6 protocol [17], in the base specification has defined mandatory multicast support, in contradiction to IPv4 [16] which is optional, IPV6 also has a new address autoconfiguration system [18] important for IP mobility.

A decade ago the majority of the devices with Internet connection were static, wired. Today most devices are wireless, WiFi or mobile (GPRS, UMTS, LTE). The use of mobile systems like laptops, tablets or smartphones have produced a fast growing in mobile communications.

Thanks to several studies by the CNMC [19] [20] [21] (Comisión Nacional del Mercado de las Comunicaciones) we can see the evolution of the wireless (3G and 4G) coverage map in Spain:

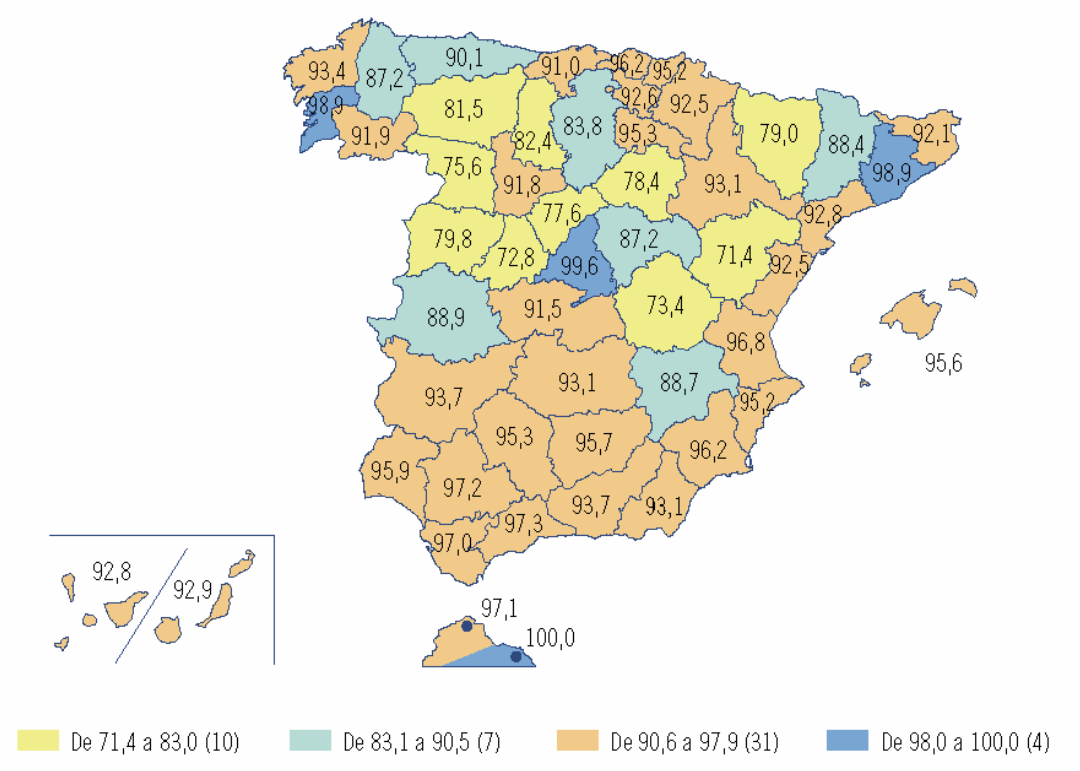


Ilustración 5: 3G coverage map of Spain in 2011 [19]

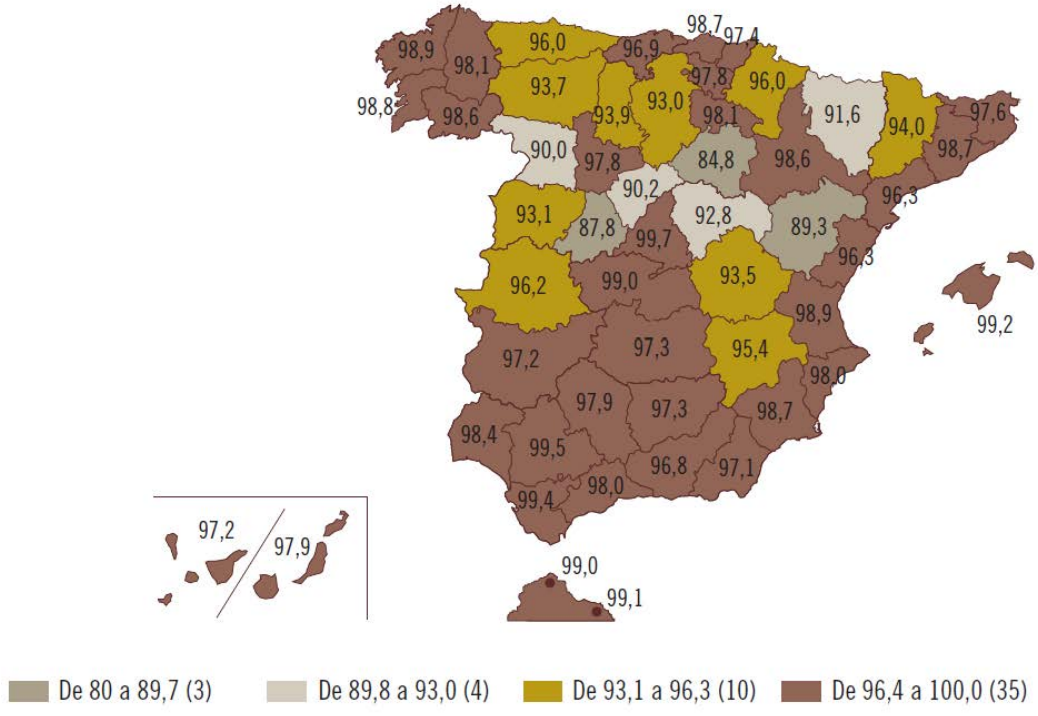


Ilustración 6: 3G coverage map of Spain in 2013 [20]

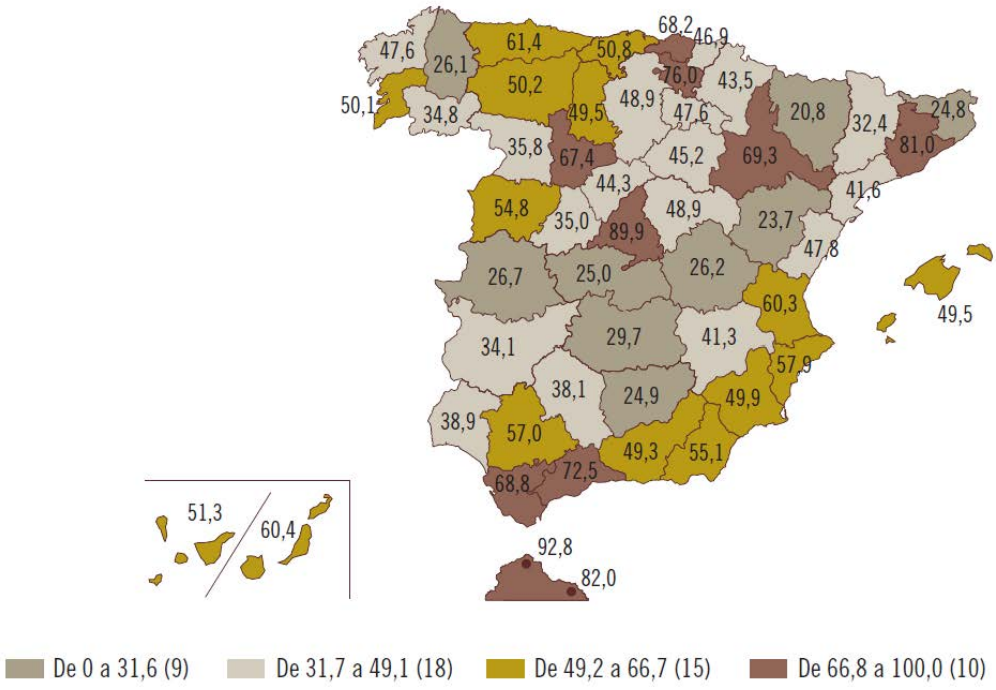


Ilustración 7: 4G coverage map of Spain in 2013 [20]

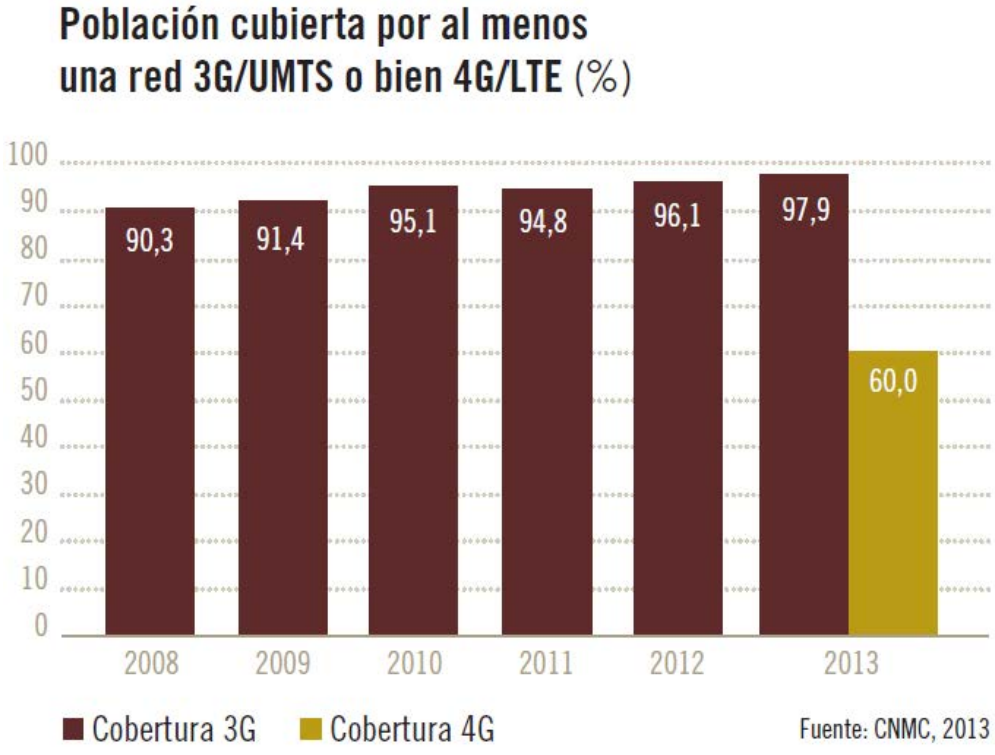


Ilustración 8: 3G and 4G coverage evolution in Spain [21]

As we can see in the graphics the growth of the mobile networks like UMTS and LTE is very high, having almost 100% of coverage in many places in Spain.

The great boom of the smartphones has led this growth, having the advantage of maintain the connection while we are moving. All of this is possible thanks to different mobility protocols that manage the movement between an access point to another maintaining the connection. Within these mobility protocols there are the network-based mobility protocols, that manage all the mobility in a transparent way to the host while it is moving. As an example of this type of protocols we have PMIPv6, the main protocol of this Final Bachelor Thesis, and GTP conceptually very similar to PMIPv6 in the mobility management. GTP is the most currently used.

A big portion of the mobile network traffic is video, a CISCO study [1] reveals that in 2014 the video traffic covered the 55% of the world traffic and their previsions growth to 72% in 2019. This video consumption is produced by the popularity of the streaming and video platforms like YouTube, the social networks, video on demand (VoD) and IPTV.

Now in Spain telecommunications corporations like Movistar has a IP television system [22] that works through multicast, a data transmission method to multiple networks and receivers simultaneously.

Sending multicast traffic, deepening in the video transmission, combined with the use of a network-based mobility protocol like PMIPv6 and its utility in UMTS/LTE networks. The combination of all of these factors, their continuous growth and the possibility of analyse and improve of these components have allowed the realization of this Final Bachelor Project.

2.3. Objectives

The main objective of this Final Bachelor Project is evaluate the transmission and reception of multicast traffic through a network-based mobility protocol to improve the performance. To reach this objective we complete different activities that can be divided in different phases:

- Study of IPv6 [17], PMIPv6 [3] and multicast (MLDv2 [4]).
- Familiarization with the OpenWRT [5] distribution. The creation of a distribution with the buildroot [23] [24] with all the requirements and the verification of the correct operation of the routers with the distribution.
- Run PMIPv6 in the router distribution. To reach this objective we have crosscompiled [8] the PMIPv6 code and we have created a basic scenario formed by three routers and two computers to run it.
- Adding IPv6 multicast features to the OpenWRT routers. We have used a program called mcproxy [9] and we have crosscompiled it to run in the OpenWRT distribution.
- Run simultaneously mcproxy and PMIPv6 in the routers.
- Evaluation of the multicast data transmission while the mobile node is performing a handoff.
- RFC 6636 [11] implementation to improve the handoff in the scenario.
- Qualitative and quantitative evaluation of the RFC 6636 and comparison between the old implementation and the RFC 6636 [11].

3. Estado del Arte

En este capítulo se van a describir las tecnologías y protocolos empleados para realizar este Trabajo Fin de Grado. En el primer apartado se procederá con una breve introducción a IPv6. En el segundo apartado se describirán los protocolos de movilidad IPv6: MIPv6 y PMIPv6. En el tercer apartado se explicará el funcionamiento de Multicast y el protocolo MLDv2.

3.1. Introducción a IPv6

Con el gran auge de Internet se ha producido un aumento de las máquinas conectadas y esto ha dado lugar al agotamiento del espacio de direcciones disponible en IPv4 [12]. Aunque existen diferentes mecanismos y herramientas, como pueden ser NAT [14] y DHCP [13], que ayudan a solventar este problema, la solución óptima es el uso de la siguiente versión llamada IPv6.

IPv6 [17] está detallado en la RFC 2460, se incluyeron diferentes mejoras respecto a su antecesor, entre las cuales está incluido el aumento del espacio de direcciones de 32 bits en IPv4 a 128 bits, lo que se traduce en 2^{128} direcciones, aproximadamente 340 sextillones de direcciones, dejando por el momento ese problema aparcado.

La cabecera ha sido rediseñada aumentando el tamaño de 20 a 40 Bytes. Varios campos de la cabecera IPv4 han sido eliminados o hechos opcionales para reducir el tiempo de procesamiento de los paquetes y para limitar el coste del ancho de banda de la cabecera IPv6. La nueva cabecera IPv6 se ha diseñado con mucha flexibilidad y con la posibilidad de introducir nuevas opciones en el futuro.

Finalmente el formato de la cabecera IPv6 queda definido de esta manera:

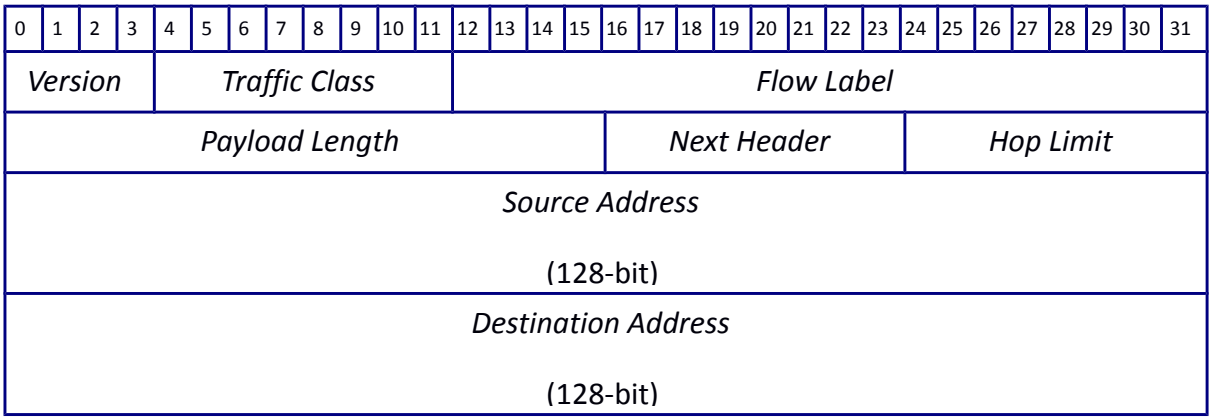


Ilustración 9: Cabecera IPv6

A continuación se van describir los diferentes campos que forman la cabecera IPv6:

- *Version* (Versión): ocupa 4 bits y muestra la versión del protocolo, en este caso 6.
- *Traffic Class* (Clase de Tráfico): ocupa 8 bits e indica la clase de tráfico, se utiliza para distinguir entre diferentes clases y prioridades en los paquetes IPv6.
- *Flow Label* (Etiqueta de flujo): ocupa 20 bits. Este campo de la cabecera sirve para etiquetar flujos de paquetes que puedan necesitar un manejo especial como no utilizar calidad de servicio o servicio en tiempo real. Los host o routers que no soporten las funciones de Flow Label pondrán este campo a 0 cuando originen un paquete y lo ignorarán cuando reciban un paquete.
- *Payload Length* (Longitud del paquete): ocupa 16 bits y especifica el tamaño total del paquete (cabecera y datos) en bytes.
- *Next Header* (Siguiente cabecera): este campo ocupa 8 bits, indica el tipo de cabecera posterior a la cabecera fija de IPv6.
- *Hop limit* (Límite de saltos): ocupa 8 bits. Es el número máximo de saltos que soportará el paquete, cuando este pase por un nodo se decrementará en una unidad, cuando llegue a cero el paquete será descartado.
- *Source Address* (Dirección Origen): ocupa 128 bits y es la dirección origen del paquete.
- *Destination Address* (Dirección Destino): ocupa 128 bits y es la dirección a la que va destinado el paquete.

IPv6 define una nueva arquitectura de direcciones [25]. Incluye los diferentes formatos para los tipos de direcciones IPV6 (unicast, anycast y multicast).

- Direcciones unicast: es un identificador de una única interfaz. Los paquetes que se envían a una dirección unicast serán entregados a esa única interfaz.
- Direcciones anycast: es un identificador para un conjunto de interfaces (pueden pertenecer a diferentes nodos). Un paquete enviado a una dirección anycast es entregado a sólo una de las interfaces del conjunto identificado por la dirección.
- Direcciones multicast: es un identificador de un conjunto de interfaces (pueden pertenecer a diferentes nodos) pero a diferencia de las direcciones anycast el paquete enviado a una dirección multicast será entregado a todas las interfaces que identifican esa dirección.

Otras características destacables de IPv6 son el descubrimiento de vecinos y la autoconfiguración de direcciones.

IPv6 incluye un sistema de descubrimiento de vecinos (ND, *Neighbor Discovery* [26]), que tiene una función similar a la que realiza ARP en IPv4. Este protocolo es el mecanismo utilizado para que los nodos que entran a una red puedan conocer el estado del enlace y la presencia de otros nodos. Para ello ND utiliza el intercambio de mensajes ICMPv6. Hay dos tipos de mensajes ICMPv6 que serán útiles para este Trabajo Fin de Grado:

- *Router Solicitation* (RS): mensaje enviado de un host a un router para generar un *Router Advertisement*.
- *Router Advertisement* (RA): mensaje que envían routers a hosts de manera periódica o mediante petición con un RS. Este mensaje contiene entre otras cosas prefijos de red que pueden ser utilizados para configuración de direcciones.

Gracias a estos mensajes ICMPv6 un router puede descubrir a sus vecinos y configurarse a si mismo una dirección IPv6. Un host cuando entra en una red espera a la recepción de un RA (ya sea por el envío periódico del router o por respuesta a un RS previamente enviado), este RA contiene información para la autoconfiguración como el prefijo de red. Una vez recibido el RA, el propio host puede autoconfigurar [18] su dirección Global unicast. Esta dirección Global se creará combinando dos partes:

1) Prefijo de red obtenido del RA.
2) Dirección de interfaz creada mediante EUI-64 (la dirección de interfaz se creará en base a la MAC del host mediante un algoritmo).

La unión de estas dos partes formará la dirección IPv6 autoconfigurada.

En este trabajo fin de grado toda la configuración del escenario se realizará mediante IPv6, y cabe destacar de las mejoras disponibles respecto a IPv4 ciertos detalles como el soporte nativo de multicast, el sistema de autoconfiguración y un mejor soporte para movilidad que son esenciales para el desarrollo del proyecto.

3.2. Movilidad IPv6: Mobile IPv6 (MIPv6) y Proxy Mobile IPv6 (PMIPv6)

El protocolo en el cual se centra este Trabajo Fin de Grado es Proxy Mobile IPv6, su funcionamiento esta basado en Mobile IPv6. Para mejorar la comprensión del protocolo central del estudio se va a realizar una breve introducción del protocolo MIPv6 y a continuación PMIPv6.

3.2.1. Mobile IPv6 (MIPv6)

Como su propio nombre indica Mobile IPv6 [2] es un protocolo de movilidad. La gran ventaja que proporciona MIPv6 es la alcanzabilidad del MN dentro y fuera de su red hogar. Cuando un MN entra por primera vez en una red MIPv6 se le asigna una dirección IP unicast permanente, esta dirección está asociada a su red hogar y se llama *Home Address* (HoA). Mientras el MN esté en su red hogar el enrutamiento de los paquetes destinados a su HoA se realizará mediante los mecanismos convencionales.

Cuando el MN esté en una red foránea adquirirá una dirección IP llamada *Care-of Address* (CoA) mediante un mecanismo de configuración IPv6 convencional, esta será la dirección temporal a la que se redirigirán los paquetes destinados a su HoA.

Para mostrar el funcionamiento básico del protocolo MIPv6 de una manera más simple y detallada se van a analizar dos escenarios:

- **Escenario 1:** este es el escenario básico en el que el MN está conectado a su red hogar. Aquí el MN configurará una HoA perteneciente al espacio de direcciones de la red hogar y esta quedará enlazada a su router en su red hogar, el *Home Agent* (HA) o agente local. El enrutamiento de paquetes se hará mediante los mecanismos convencionales a través del HA.

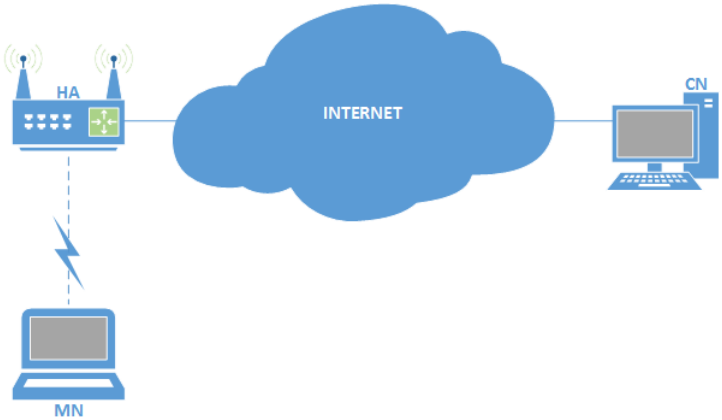


Ilustración 10: Escenario MIPv6 en red hogar

- **Escenario 2:** aquí el MN está fuera de su red hogar conectado a un Router de Acceso o *Access Router* (RA). En este escenario cobra importancia la labor del HA, cuando un MN sale de su red hogar los paquetes van destinados a su dirección permanente (HoA), el HA los intercepta para reenviarlos a la dirección temporal (CoA) que tenga el MN en ese momento.

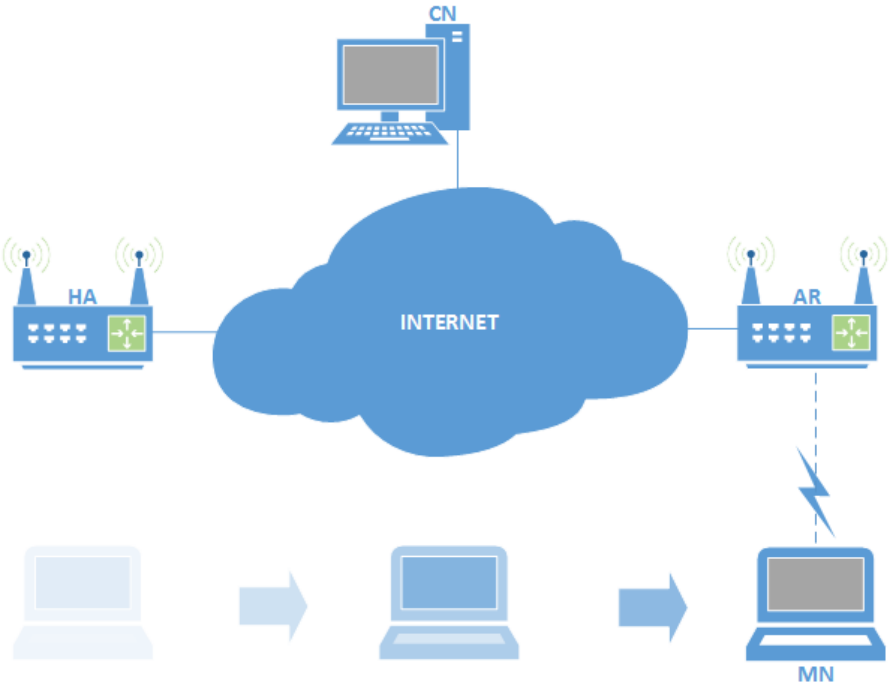


Ilustración 11: Movimiento de MN en escenario MIPv6

Una vez el MN entra en la red foránea configurará mediante los mecanismos convencionales su CoA para poder recibir los paquetes del HA. Esta CoA se asociará a la HoA en una estructura de datos en el HA, llamada *Binding Cache* (BC). La *Binding Cache* está llena de un tipo de datos llamado *Binding Cache Entry* (BCE) donde se guardan los datos necesarios para gestionar la movilidad. Para poder almacenar los datos en una BCE se realiza un intercambio de mensajes entre el MN y el HA. El MN mandará un *Binding Update* (BU) y el HA le confirmará la actualización de los datos con un *Binding Acknoledgement* (BA).

Cada vez que un Nodo Corresponsal o *Correspondent Node* (CN) envíe un paquete con destino la HoA del MN será interceptado por el HA, lo encapsulará y enviará a través de un túnel bidireccional a la CoA del MN gracias a la asociación producida en la BC. El funcionamiento inverso será de la misma manera, el envío de paquetes del MN al CN se encapsulará en un túnel hacia el HA y de aquí al CN.

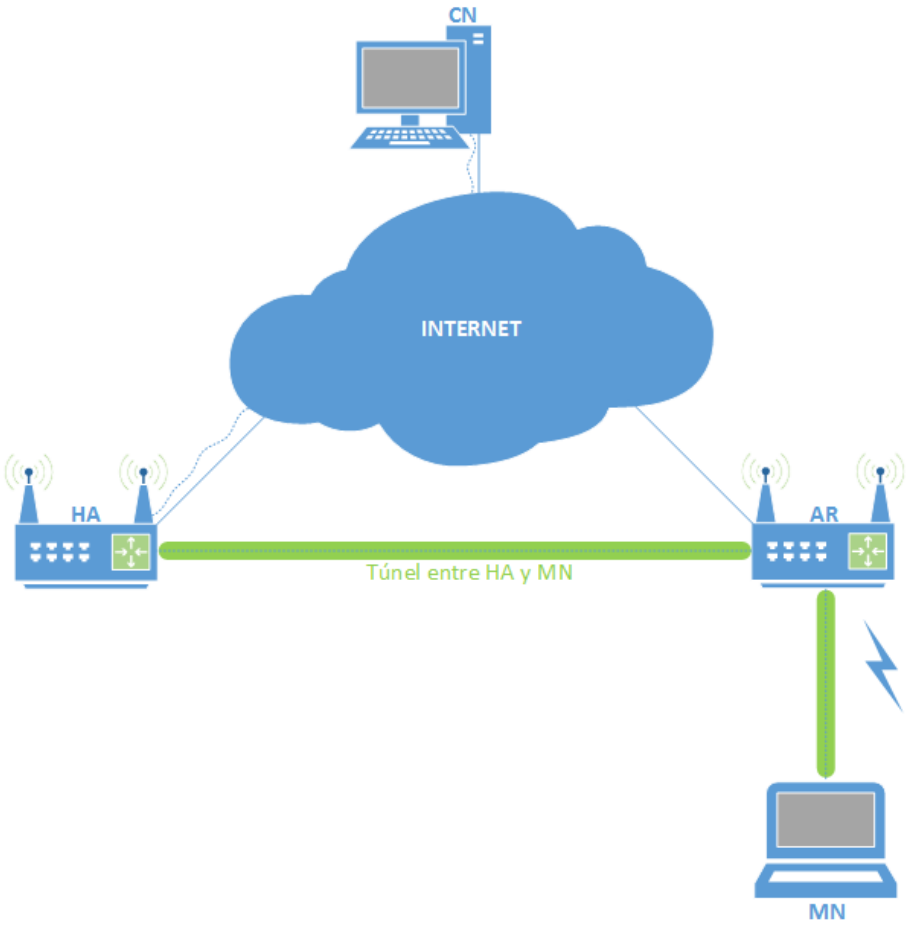


Ilustración 12: Túnel entre HA y MN en MIPv6

La encapsulación de un paquete consiste en añadir la nueva cabecera IP e introducir la cabecera anterior en el payload. En la recepción del paquete encapsulado se eliminará la nueva cabecera añadida quedando el paquete intacto.

Una vez el MN abandona la red foránea para volver a la red hogar se mandará un BU y recibirá el correspondiente BA para eliminar la CoA de la BC y seguidamente eliminar el túnel bidireccional volviendo de nuevo al escenario 1.

3.2.2. Proxy Mobile IPv6 (PMIPv6)

Proxy Mobile IPv6 [3] es el protocolo principal de este Trabajo Fin de Grado. PMIPv6 es un protocolo de gestión de movilidad basado en red y está basado en MIPv6, por lo que tiene un funcionamiento y terminología similar, pero a diferencia de este, ofrece movilidad a un nodo sin que éste participe en la señalización de movilidad IP. Existen dos entidades de movilidad:

- *Mobile Access Gateway* (MAG): es el router de acceso del nodo móvil, gestiona toda la señalización de movilidad. Es el responsable del seguimiento de la movilidad del nodo móvil y de su señalización al LMA.
- *Local Mobility Anchor* (LMA): es el equivalente al HA de MIPv6. Esta entidad se encarga de mantener todas las rutas para cada MN del dominio PMIPv6. Topológicamente es el punto de anclaje de las direcciones asignadas a los MN, esto significa que todos los paquetes destinados a estas direcciones serán enrutados al LMA.

Hay dos situaciones que cabe destacar a lo largo de la estancia de un MN en un dominio PMIPv6: cuando el MN entra por primera vez a un dominio PMIPv6 y se conecta a un punto de acceso (MAG) y cuando se mueve dentro del dominio PMIPv6 y cambia de MAG (acción llamada *handoff*).

Para analizar estas situaciones vamos a mostrar un escenario básico, como el utilizado a lo largo del Trabajo Fin de Grado, formado por un LMA, dos MAG, un MN y un CN.

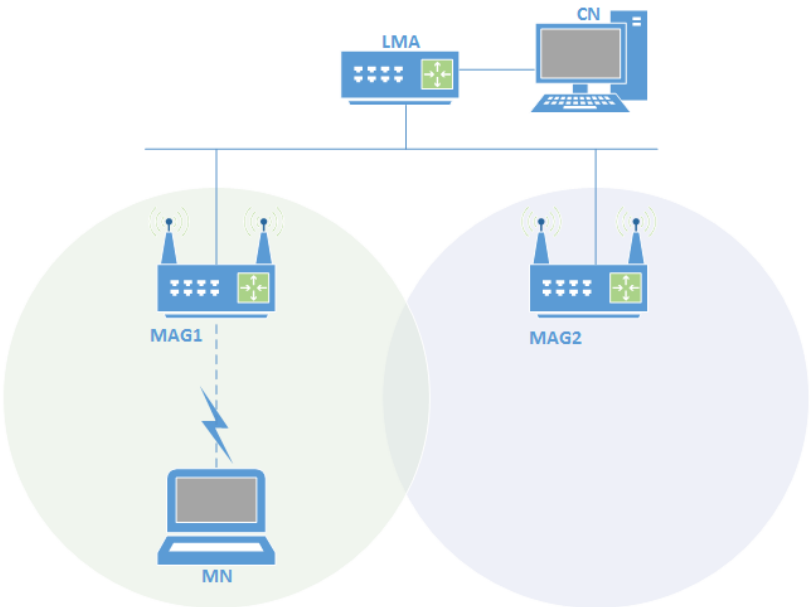


Ilustración 13: Escenario PMIPv6

Primeramente vamos a explicar de una manera sencilla la primera situación, en la que el MN entra por primera vez a un dominio PMIPv6 y se conecta a un punto de acceso (MAG1). Cuando el MN se intenta conectar a MAG1, este identifica al MN y determina si está autorizado para un servicio de gestión de movilidad basada en red. Si la red determina que el MN está autorizado entonces mediante mecanismos de autoconfiguración IPv6 el MN configurará una dirección IPv6 perteneciente al espacio de direcciones del *Home Network Prefix* (HNP) o prefijo de red hogar. Con esa dirección configurada en su interfaz el MN podrá moverse a través del dominio PMIPv6 de manera transparente y manteniendo esa misma dirección basada en el HNP.

A continuación vamos a ver de manera más detallada la primera situación, se van a mostrar todos los mensajes intercambiados entre el MN, el MAG1 y el LMA y todos los pasos seguidos hasta la autoconfiguración de la dirección IPv6 perteneciente al HNP en el MN:

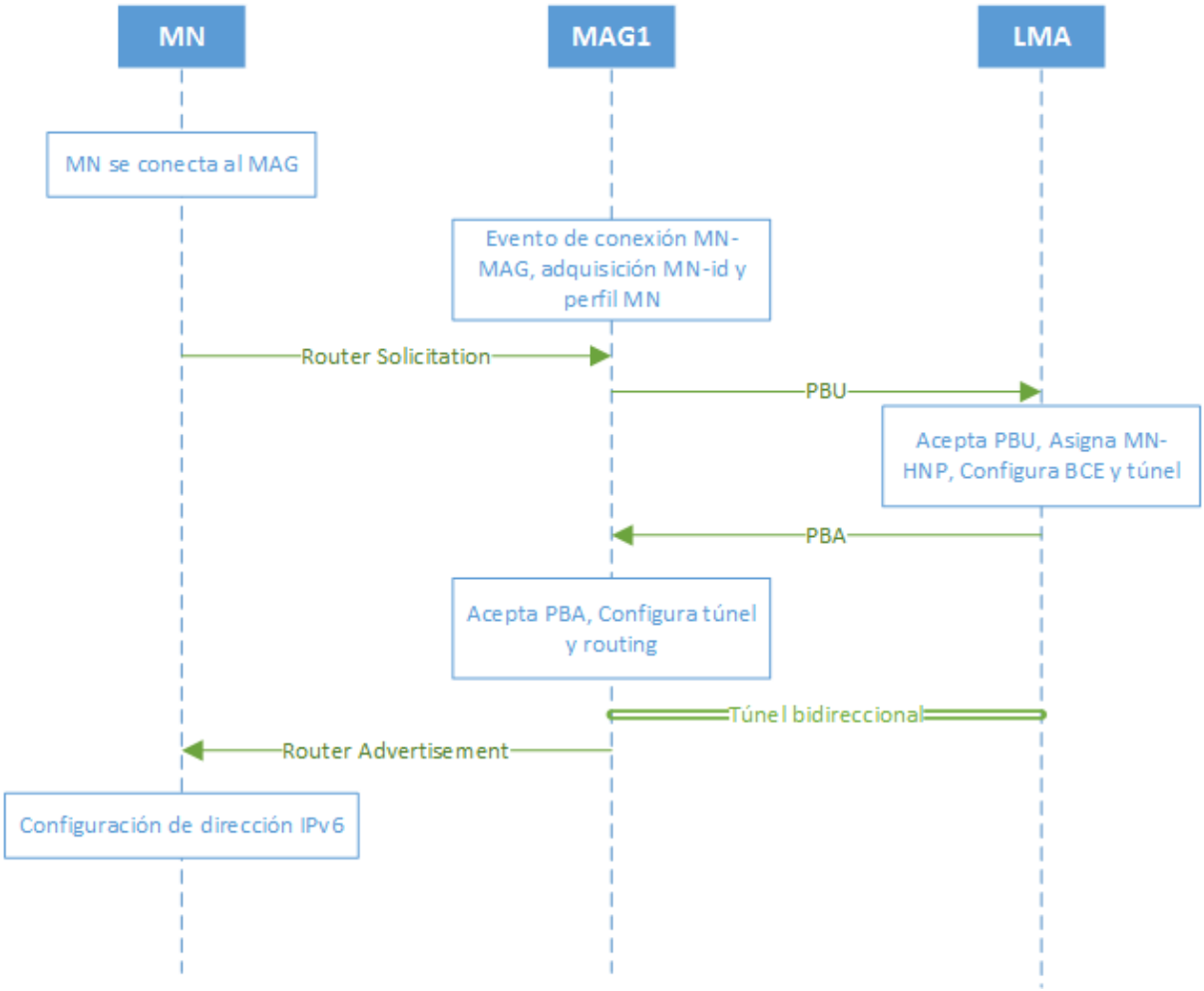


Ilustración 14: Señalización PMIPv6

La ilustración 14 muestra el intercambio de mensajes en la entrada del MN en el dominio PMIPv6. En algún momento posterior a la entrada del MN en el dominio PMIPv6 el MN envía un *Router Solicitation* al MAG1, una vez este es recibido por el MAG1 manda un mensaje llamado *Proxy Binding Update* (PBU), el objetivo de este mensaje es solicitar la creación de un enlace entre el HNP del MN y la dirección del MAG. El LMA recibe el PBU y procede a crear una *Binding Cache Entry* (BCE) del MN, configura su punto de enlace del túnel bidireccional y responde el PBU con un PBA, que contendrá la información correspondiente a HNP que deberá configurar el MN. El LMA al recibir este PBA configura su punto de enlace del túnel bidireccional y configura el reenvío de tráfico al MN. MAG1 ya está listo para mandar RA al MN para que este configure la IPv6 perteneciente al prefijo de red del dominio PMIPv6.

Ahora el LMA como punto de enlace del prefijo de red hogar, recibirá todo el tráfico de cualquier nodo fuera del dominio PMIPv6 (CN) dirigido al MN y lo reenviará y encapsulará a través del túnel bidireccional hacia el MAG. El MAG1 eliminará la cabecera introducida en el túnel y entregará al MN el paquete original.

Ahora vamos a centrarnos en la siguiente situación, el *handoff*, cuando el MN se mueve a través del dominio PMIPv6 y cambia de MAG.

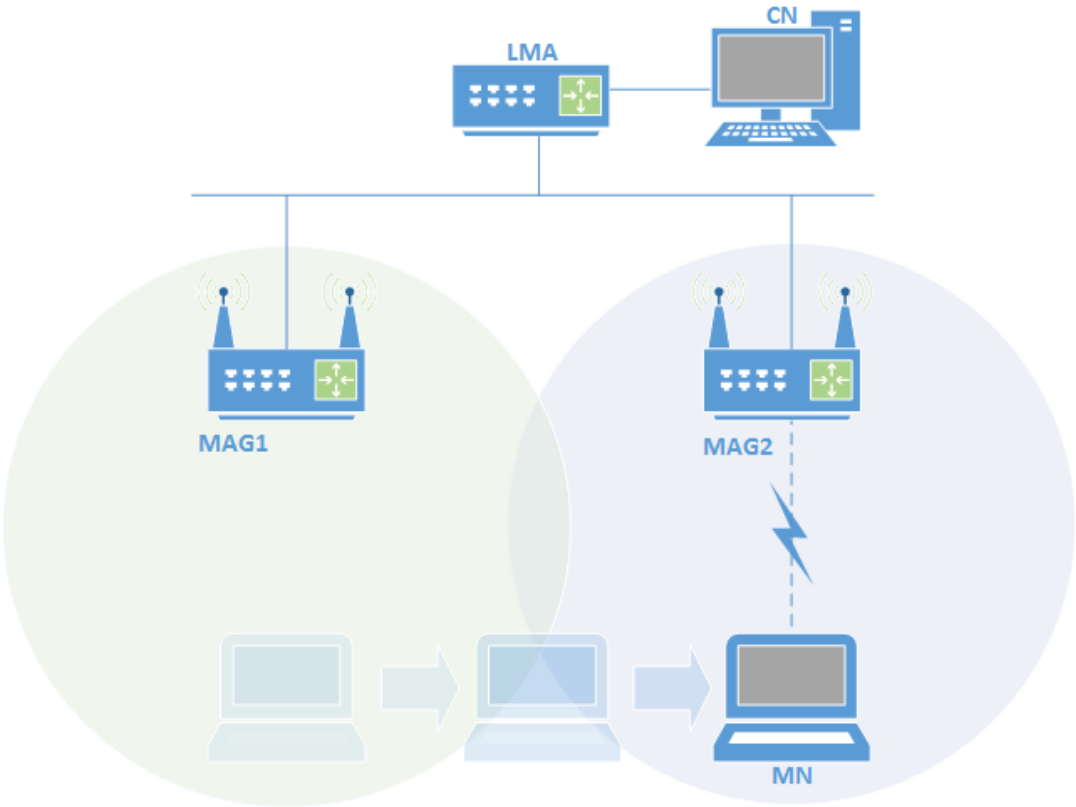


Ilustración 15: Cambio de MAG en PMIPv6

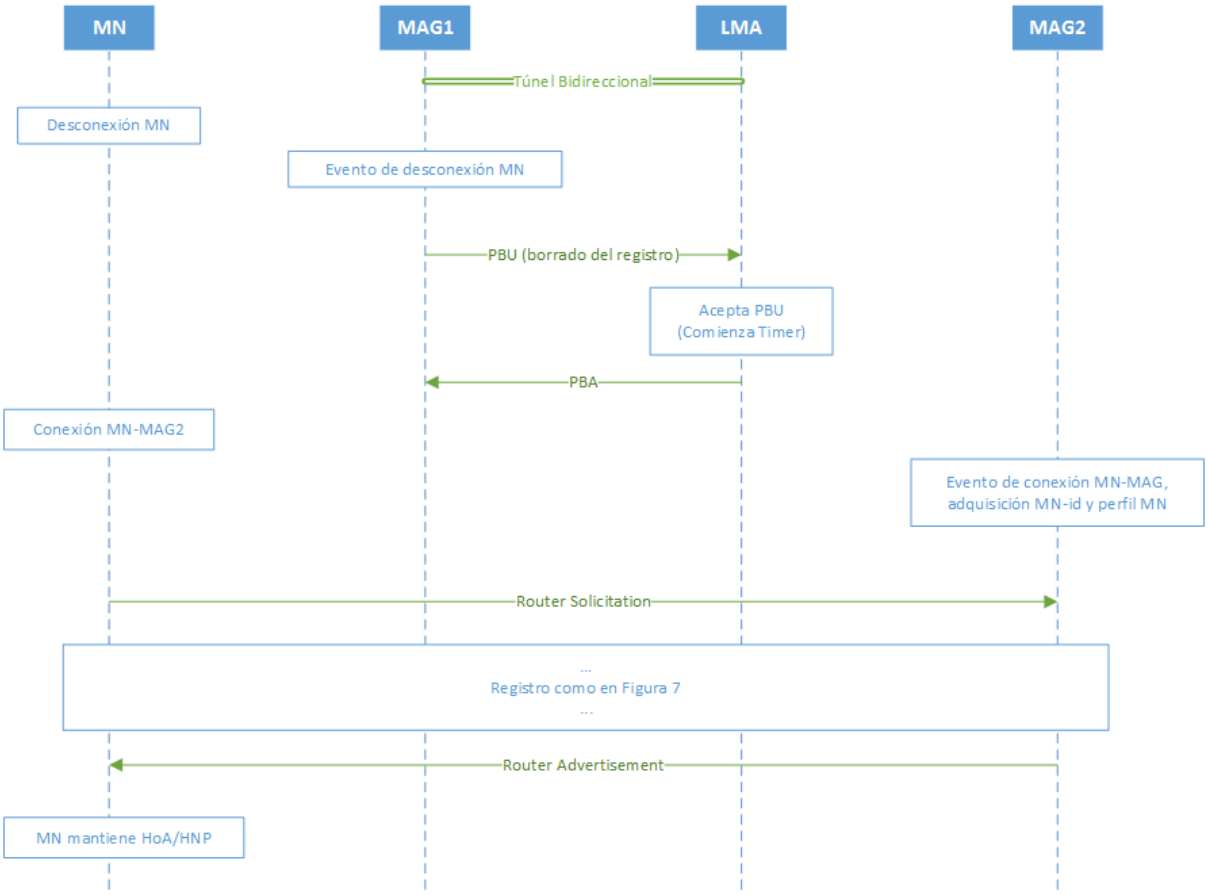


Ilustración 16: Señalización cambio de MAG en PMIPv6

Como podemos ver en la ilustración 16 está detallado el intercambio de mensajes necesario para que el nodo móvil haga un *handoff* de MAG1 a MAG2. Cuando el MN se mueve a MAG2 MAG1 lo detecta y comienza un proceso de eliminado del *binding* y mandará un PBU al LMA. El LMA al recibir el PBU identifica la sesión del MN y esperará un periodo de tiempo para que el MAG2 actualice su información y envíe un PBU que anuncie que el MN se ha movido a MAG2. En caso de no recibir ningún PBU a la conclusión del *timer,* el LMA borrará la BCE del MN. Cuando el MAG2 detecte al MN se lo señalizará al LMA para que actualice su BCE, todo se realizará mediante un proceso igual al mostrado en la ilustración 14. Una vez terminada la señalización, el MN recibirá un RA con el mismo MN-HNP que antes y el MN no notará ningún cambio a nivel IP en MAG2 respecto a MAG1.

La línea de mensajes en ilustración 16 no tiene que seguir estrictamente ese orden lógico ya que puede que el MAG2 notifique al LMA antes la localización del MN en su red que MAG1 el abandono de la suya.

3.3. Multicast

3.3.1. Multicast en IPv6

Para comenzar vamos a dar una breve explicación de qué es multicast. Multicast o multidifusión es un método de transmisión de paquetes a un grupo de receptores, este envío se realiza mediante la suscripción de los receptores a una IP multicast.

En IPv6 se ha introducido soporte nativo para multicast. Dentro del espacio de direcciones de IPv6 hay un rango reservado para multicast, definido en el punto 2.7 de la RFC 4291 [25], a continuación se va a detallar el formato de este tipo de direcciones en IPv6:

8	4	4	112 bits
11111111	*flags*	*scope*	*Group ID*

Ilustración 17: Estructura de una dirección Multicast IPv6

Los 8 bits iniciales a 1 identificarán esta dirección como una dirección IPv6 multicast.

Scope define el alcance del grupo multicast:

0	Reservado	8	*Organization-Local scope*
1	*Interface-Local scope*	9	(Sin asignar)
2	*Link-Local scope*	A	(Sin asignar)
3	Reservado	B	(Sin asignar)
4	*Admin-Local scope*	C	(Sin asignar)
5	*Site-Local scope*	D	(Sin asignar)
6	(Sin asignar)	E	*Global Scope*
7	(Sin asignar)	F	Reservado

Ilustración 18: Alcance del grupo Multicast

- *Interface-Local*: abarca sólo la interfaz en un nodo y es útil únicamente para transmisión *loopback* en multicast.
- *Link-Local*: abarca la misma región topológica del alcance unicast correspondiente.
- *Admin-Local*: es el ámbito más pequeño que se debe configurar administrativamente, no se deriva automáticamente de la conectividad física o de otra configuración no relacionada con multicast.

- *Site-Local*: para abarcar un sólo site.
- *Organization-Local*: para abarcar múltiples *sites* que pertenezcan a una única organización.
- (Sin asignar) están reservadas para administradores, para definir regiones multicast adicionales.

Group ID identifica al grupo multicast, ya sea permanente o transitorio dentro del ámbito definido en *scope*.

Flags define los 4 *flags*:

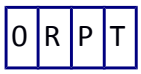


Ilustración 19: *Flags* de una dirección Multicast IPv6

- El primer *flag* está reservado y debe estar inicializado a 0.
- El *flag* T tiene dos opciones T=0 y T=1:
  - T=0 indica que es una dirección multicast permanentemente asignada("*well-known*").
  - T=1 indica que la dirección multicast no es permanentemente asignada (es transitoria o asignada dinámicamente).
- El flag P tiene dos opciones P=0 y P=1, está detallado en la RFC 3306 [27]. Para la explicación de este *flag* se desglosa de una manera más detallada el formato de la dirección multicast IPv6 para esta situación:

8	4	4	8	8	64	32
11111111	*flgs*	*scop*	*reserved*	*plen*	*network prefix*	*group ID*

Ilustración 20: Estructura de una dirección Multicast IPv6 para *flag* P

Los *flags* en este caso se dispondrán de esta manera:

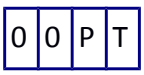


Ilustración 21: *Flags* de una dirección Multicast IPv6 para *flag* P

- P=0 indica una dirección multicast que no es asignada en función del prefijo de red.

- P=1 indica una dirección multicast que ha sido asignada en función del prefijo de red. Si P=1 entonces T debe ser 1 también.
- El campo *reserved* debe ser inicializado a 0.
- El campo plen indica el número de bits en el campo *network prefix* que identifica a la subred cuando P=1.
- El campo *network prefix* indica el prefijo de la subred unicast a la que pertenece la dirección multicast.

- El flag R está detallado en la RFC 3956 [28]. Para la explicación de este flag se desglosa de una manera más detallada el formato de la dirección multicast IPv6 para esta situación:

8	4	4	4	4	8	64	32
11111111	*flgs*	*scop*	*rsvd*	RIID	*plen*	*network prefix*	*group ID*

Ilustración 22: Estructura de una dirección Multicast IPv6 para *flag* R

- Cuando R=1 indica una dirección multicast que incluye la dirección para el *Rendezvous Point*, en este caso P debe ser 1 y T deber ser 1.
- Cuando R=0 indica una dirección multicast que no incluye la dirección para el *Rendezvous Point*. En este caso RIID debe ser 0.

3.3.2. Multicast Listener Discovery Version 2 en IPv6 (MLDv2)

MLDv2 [4] es un protocolo usado en IPv6 para descubrir la presencia de subscriptores multicast en enlaces directamente conectados y para descubrir que direcciones multicast son de interés para los nodos vecinos.

MLD es un protocolo asimétrico, especifica diferentes comportamientos para subscriptores de direcciones multicast y los routers multicast. El propósito de MLD es que los routers multicast puedan aprender de cada enlace directamente conectado en qué direcciones multicast y en qué fuentes están interesados los subscriptores multicast del enlace. En MLDv2 existen dos tipos de mensajes:

- *Multicast Listener Query*: son enviados por lo routers para consultar el estado de multicast *listening* en las interfaces vecinas. Hay tres variantes del *Query*:
  - *General Query* (*Query* general): es enviado por el *Querier* para aprender qué direcciones multicast quieren los subscriptores en el enlace.
  - *Multicast Address Specific Query* (*Query* específico de dirección multicast): es enviado por el *Querier* para aprender si una dirección multicast particular tiene algún subscriptor en el enlace.
  - *Multicast Address and Source Specific Query* (*Query* específico de dirección multicast y fuente): es enviado por el *Querier* para aprender si alguna de las fuentes de la lista especificada para la dirección multicast particular tiene algún subscriptor en el enlace.

Los *Queries* suelen enviarse a todos los nodos del enlace con la dirección multicast FF02::1. Los paquetes de *Multicast Listener Query* tienen esta estructura:

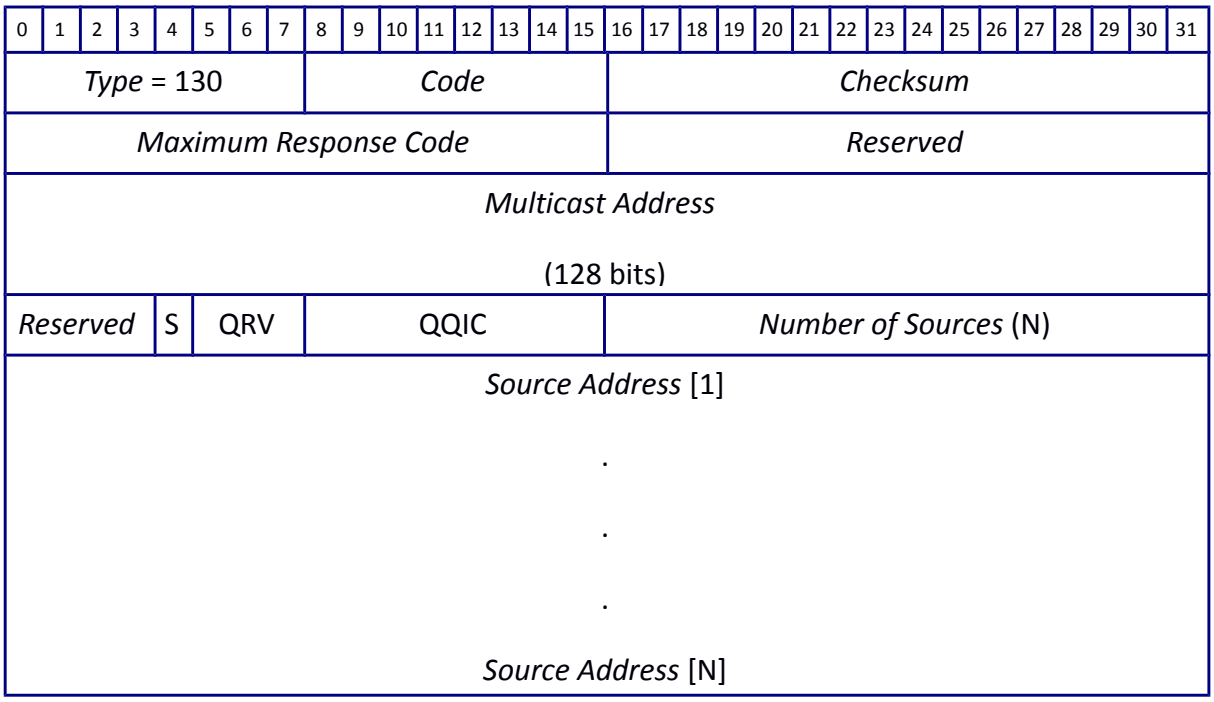


Ilustración 23: Estructura de un paquete *Multicast Listener Query*

- *Code*: inicializado a 0 por el remitente e ignorado por los receptores.
- Checksum: ICMPv6 checksum estándar. Cubre el mensaje MLDv2 completo además de unas *pseudo-cabeceras* IPv6. Para calcular el checksum el campo se pone a 0. Cuando se recibe un paquete se debe verificar el checksum antes de procesarlo.
- *Maximum Response Code*: de su valor deriva el *Maximum Response Delay* (Retardo Máximo de Respuesta), es el tiempo máximo permitido antes de enviar un *Report*.
- *Reserved*: inicializado a 0 por el remitente e ignorado por los receptores.
- *Multicast Address*: para un *Query* general este campo estará a 0. Para un *Query* específico contiene la dirección multicast consultada.
- *Flag* S: cuando este *flag* está a 1 indica a los routers multicast que reciben este mensaje que tienen que suprimir las actualizaciones normales del temporizador que realicen al escuchar un *Query*.
- QRV (*Querier's Robustness Variable*): si no es 0 este campo contiene la variable de robustez, si supera el valor de 7 (el máximo valor que soporta esta variable) se pondrá a 0.

- QQTC (*Querier's Query Interval Code*): indica el intervalo de *Query* utilizado por el *Querier*.
- *Number of Sources* (N): indica el número de direcciones presentes en el *Query*, en un *Query* general su valor es 0. Este valor está limitado por el MTU del enlace.
- *Source Address* [i]: es un vector que contiene n direcciones unicast.

- *Multicast Listener Report*: es un mensaje enviado por los nodos IP para informar a los routers vecinos de su *Multicast Listening State*. Tiene el siguiente formato:

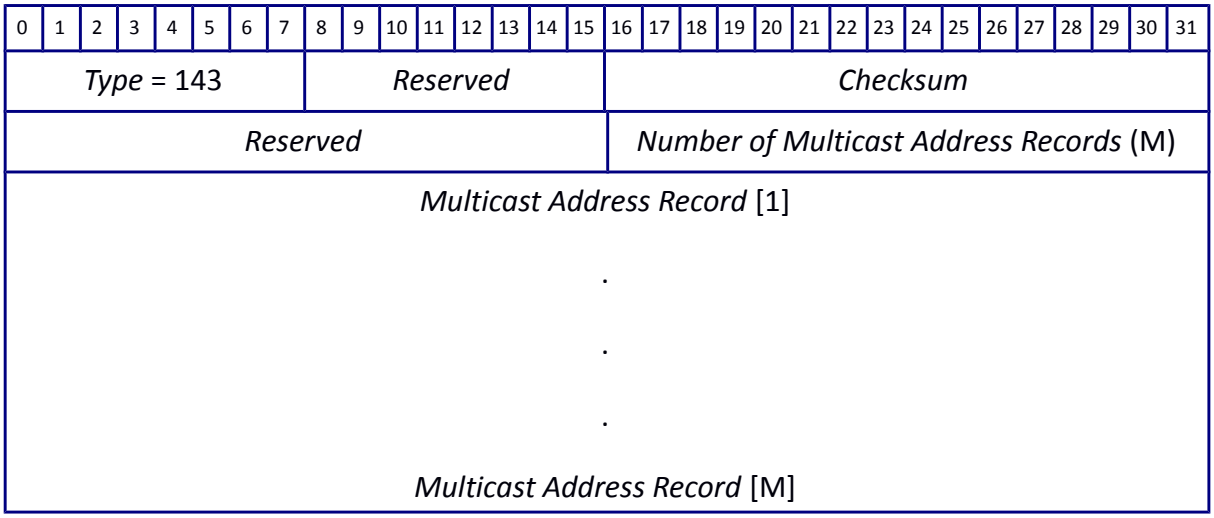


Ilustración 24: Estructura de un paquete *Multicast Listener Report*

Cada *Multicast Address Record* tiene el siguiente formato:

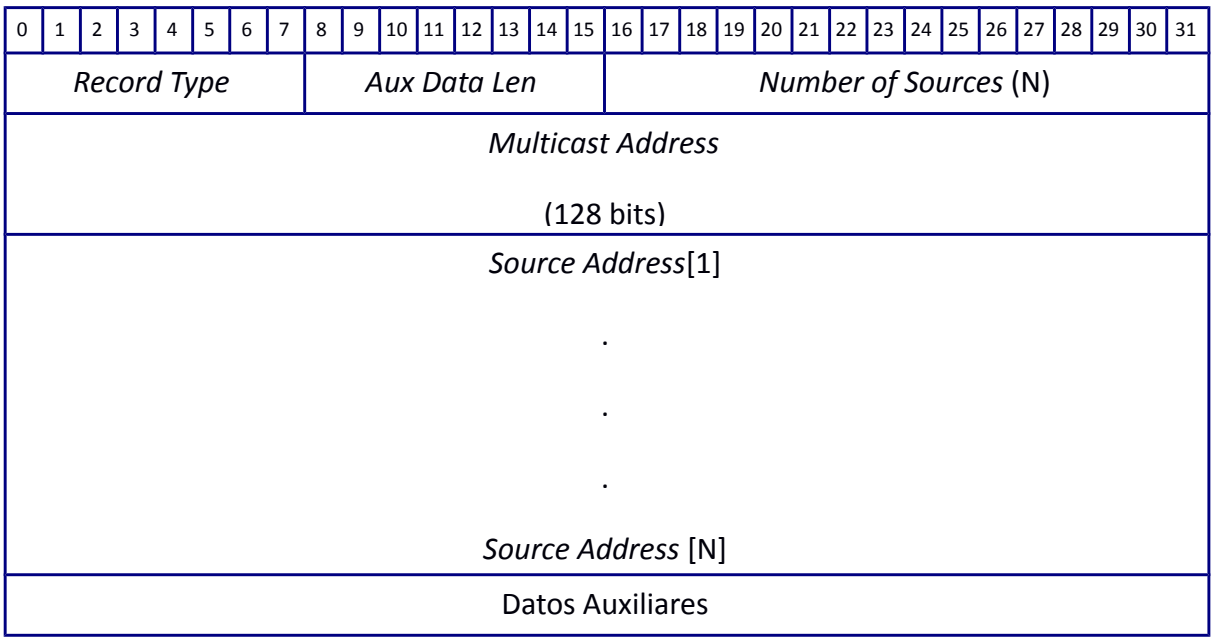


Ilustración 25: Estructura de un *Multicast Address Record*

- *Reserved*: este campo será puesto a 0 en la transmisión e ignorado en la recepción.
- *Checksum*: ICMPv6 checksum estándar. Cubre el mensaje MLDv2 completo además de unas *pseudo-cabeceras* IPv6. Para calcular el checksum el campo se pone a 0. Cuando se recibe un paquete se debe verificar el checksum antes de procesarlo.
- *Number of Multicast Address Records* (M): indica el número de *Multicast Address Records* incluidos en el *Report*.
- *Multicast Address Record*: cada *Multicast Address Record* es un bloque de diferentes campos que contiene información sobre el subscriptor remitente. Está formado por los siguientes campos:
  - *Record Type*: especifica el tipo de *Multicast Address Record*.
  - *Aux Data Len*: contiene la longitud de los datos auxiliares. Si su valor es 0 significa que no contiene datos auxiliares.
  - *Number of Sources* (N): indica el número de direcciones en el *Multicast Address Record*.
  - *Multicast Address*: indica la dirección a la cual pertenece el *Multicast Address Record*.

- *Source Address* [i]: vector de direcciones unicast.
- Datos auxiliares: contiene información adicional.

En MLDv2 se especifican diferentes temporizadores para gestionar las suscripciones multicast, estos temporizadores son modificables. A continuación se detallan cada uno de ellos:

- *Robustness Variable*: permite ajustar su valor a la pérdida de paquetes en un enlace. Si el enlace tiene riesgo de pérdida se puede aumentar su valor. MLD es robusto siempre que el número de paquetes perdidos no supere Robustness Variable – 1. Nunca debe valer 0 y su valor por defecto es 2.
- *Query Interval*: es el intervalo de envío entre *General Queries* enviados por el *Querier*. Su valor por defecto es 125 segundos.
- *Query Response Interval*: es el *Maximum Response Delay* utilizado para calcular el *Maximum Response Code* de los *General Queries* periódicos. Su valor por defecto es 10000 milisegundos. Siempre debe ser menor al *Query Interval*.
- *Multicast Address Listening Interval*: es el tiempo que debe pasar antes de que un router multicast decida que no hay más subscriptores para una dirección multicast o una fuente concreta en un enlace. Su valor debe ser (*Robustness Variable*)*(*Query Interval*)+(*Query Response Interval*).
- *Other Querier Present Timeout*: es la cantidad de tiempo que debe transcurrir antes de que un router multicast decida que otro router multicast ya no deber ser el *Querier*. Su valor debe ser (*Robustness Variable*)*(*Query Interval*)+0.5*(*Query Response Interval*).
- *Startup Query Interval*: es el intervalo de tiempo entre el envío de *General Queries* por un *Querier* en el arranque. Su valor por defecto es ¼ del *Query Interval*.
- *Startup Query Count*: es el número de *Queries* mandados en el arranque, separados por el *Startup Query Interval*. Su valor por defecto es el valor de *Robustness Variable*.
- *Last Listener Query Interval*: es el *Maximum Response Delay* usado para calcular el *Maximum Response Code* que está presente en los *Multicast Address Specific Queries* enviados en respuesta a un mensaje *Multicast Listener Done* Versión 1. Es también el *Maximum Response Delay* que se utiliza para calcular el *Maximum Response Code* insertado en *Multicast Address and Source Specific Queries*. Su valor por defecto 1000 milisegundos.

- *Last Listener Query Count*: es el número máximo de *Multicast Address Specific Queries* mandados antes de que el router asuma que no hay subscriptores locales. También es el número máximo de *Multicast Address and Source Specific Queries* enviados antes de que el router asuma que no hay subscriptores para una fuente particular. Su valor por defecto es *Robustness Variable.*
- *Last Listener Query Time*: (*Last Listener Query Interval*)*(*Last Listener Query Count*).
- *Unsolicited Report Interval*: es el tiempo entre repeticiones de un informe de interés inicial de un nodo en una dirección multicast. Su valor por defecto es 1.
- *Older Version Querier Present Timeout*: es un temporizador de compatibilidad con MLDv1.
- *Older Version Host Present Timeout*: es un temporizador de compatibilidad con MLDv1.

4. Trabajo realizado

4.1. Introducción

En este apartado de la memoria se describirán las distintas fases realizadas en el desarrollo de este Trabajo Fin de Grado. Con este proyecto se busca evaluar mecanismos de soporte de tráfico multicast con movilidad basada en red, para poder realizar esta evaluación ha sido fundamental definir un escenario donde realizar las pruebas. Este escenario estará formado por tres routers (LMA, MAG1 y MAG2) dotados de diferentes funcionalidades y dos ordenadores (MN y CN) que se comunicarán entre sí a través de estos routers. Los tres routers funcionarán bajo un protocolo de movilidad basado en red formando un dominio PMIPv6. El CN estará conectado mediante un cable ethernet al router LMA, siendo este el transmisor del tráfico multicast en el escenario. El MN se conectará inalámbricamente al dominio PMIPv6, siendo este el receptor de todo el tráfico enviado por el CN. Aquí se puede ver gráficamente el escenario junto a la asignación de direcciones IP:

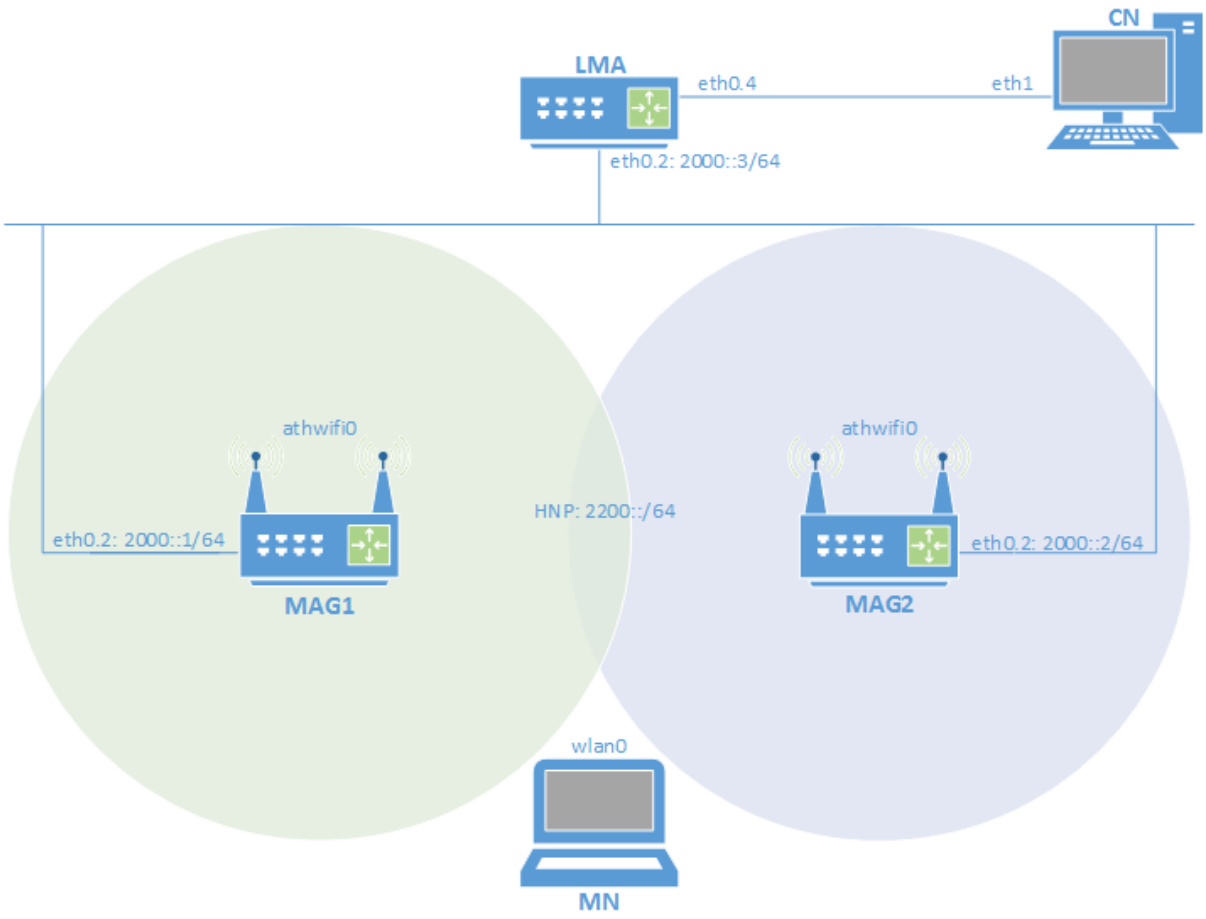


Ilustración 26: Escenario Final

4.2. Elección y configuración de los routers

En este apartado se explicará la elección del modelo de router y su configuración para formar el escenario final. Una explicación más detallada y técnica de la configuración se muestra en el anexo B OpenWRT y configuración de routers.

Inicialmente este Trabajo Fin de Grado se comenzó con el modelo de router **Linksys WRT54GL** [6] pero debido a la falta de memoria resultó imposible ejecutar simultáneamente PMIPv6 y mcproxy y fue descartado. Se decidió utilizar un modelo más actual y con mayor capacidad de hardware como es el modelo **Asus WL-500G Premium** [10]:

Ilustración 27: Asus WL-500G Premium

Especificaciones de hardware del Asus WL-500G Premium:

Versión	CPU	RAM	Flash	Network	USB	Serial	JTag
v1	Broadcom BCM4704@264Mhz	32MB	8MB	4x1	2x2.0	Sí	No

Tabla 1: Especificaciones Asus WL-500G Premium

En este modelo de router instalamos la última versión de la distribución Linux OpenWRT [5] disponible, Barrier Breaker (14.07) [7]. Sobre esta versión de OpenWRT se creará una imagen a medida añadiendo y eliminando las funcionalidades necesarias para hacer una imagen completa y lo más ligera posible.

Para crear esta imagen personalizada se ha utilizado la herramienta de compilación *buildroot* de OpenWRT [23] [24]. Gracias a esta herramienta podemos modificar la imagen que será instalada en el router, eliminar funcionalidades innecesarias para nuestro uso o

añadir funcionalidades esenciales. Una vez diseñada la imagen final se procederá a su compilación para su posterior instalación. Toda la información a cerca de la creación, compilación y instalación de imágenes OpenWRT viene detallada en el anexo B. OpenWRT y configuración de Router.

Una vez creada la imagen personalizada e instalada en cada uno de los tres routers se procederá a la configuración de todas las interfaces de manera acorde al escenario final mostrado en la ilustración 26. En estas interfaces se hará una asignación IPv6 para la comunicación del protocolo PMIPv6, y a su vez una asignación IPv4 para una conexión telnet que facilite su configuración remota a través de la red.

Se pueden ver la totalidad de los comandos utilizados en los pasos de la configuración en el anexo B. OpenWRT y configuración de Router.

4.3. Instalación y prueba de PMIPv6

Una vez los routers funcionen correctamente bajo la distribución OpenWRT pasaremos al siguiente paso, la creación del escenario PMIPv6 descrito en la ilustración 26. Para ello es necesaria una implementación de PMIPv6, para este Trabajo Fin de Grado se ha utilizado una cedida por el tutor Carlos Jesús Bernardos. Esta implementación es un programa en C sin soporte para OpenWRT por lo que habrá que realizar una compilación cruzada del código para hacerlo funcional en el router, los detalles de la compilación cruzada [8] están en el anexo B.5. Compilación cruzada de código para OpenWRT.

Una vez compilado el programa se copiará el ejecutable al router y se configurará de manera acorde al anexo B.7.1. Configuración PMIPv6.

Una vez lista la configuración del router, con todas las direcciones e interfaces correctamente asignadas, junto a la configuración de PMIPv6, el escenario 19 estará listo para funcionar.

Para lanzar el programa sólo es necesario situarse en el directorio del ejecutable copiado y lanzar el programa. Cada router necesita una ejecución diferente, a continuación se muestran los tres comandos utilizados:

MAG1: **./pmip6d -m -L 2000::3 -N 2000:1::1 -E 2000::1 -i -d10**

MAG2: **./pmip6d -m -L 2000::3 -N 2000:2::1 -E 2000::2 -i -d10**

LMA: **./pmip6d -a -L 2000::3 -i -d10**

Una vez el demonio esté ejecutándose en los routers en segundo plano es momento de que el MN entre en el dominio PMIPv6. Como se ha descrito anteriormente en el estado del arte, PMIPv6 es un protocolo de movilidad basado en red, esto permite que el MN se conecte a esta red de manera transparente y sin requerir ningún tipo de implementación, una vez dentro de la red mantendrá su dirección asignada aunque cambie de MAG.

Para que el MN se conecte a la red es necesario que disponga de conexión inalámbrica. Para el manejo de la gestión inalámbrica se ha utilizado la herramienta iw. Para que el nodo móvil se conecte al MAG se ha utilizado el siguiente comando en terminal:

sudo iw wlan0 connect MAG1

Una vez el MN se conecte se intercambiarán una serie de mensajes entre el MN y el MAG y el MAG con el LMA, finalizando con la asignación del HNP definido en el archivo match. A continuación se va a ilustrar la entrada del MN en el dominio PMIPv6:

Aquí se podemos el intercambio de mensajes gracias a Wireshark, una herramienta que permite capturar el tráfico de una red. Su instalación está detallada en el anexo C.4.

- *Router Solicitation* del MN al MAG:

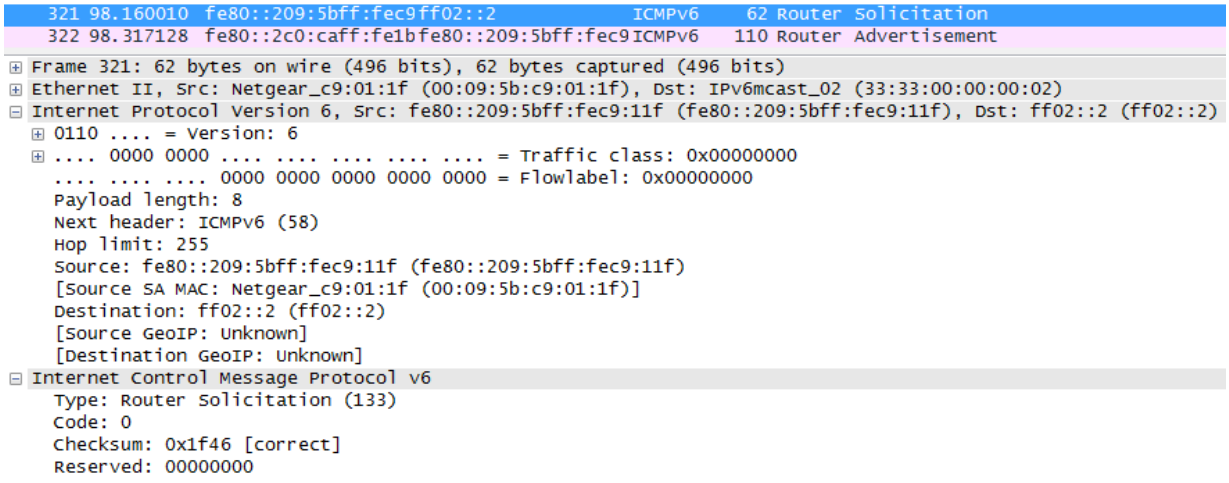


Ilustración 28: Captura de Wireshark de un *Router Solicitation*

- PBU del MAG al LMA:

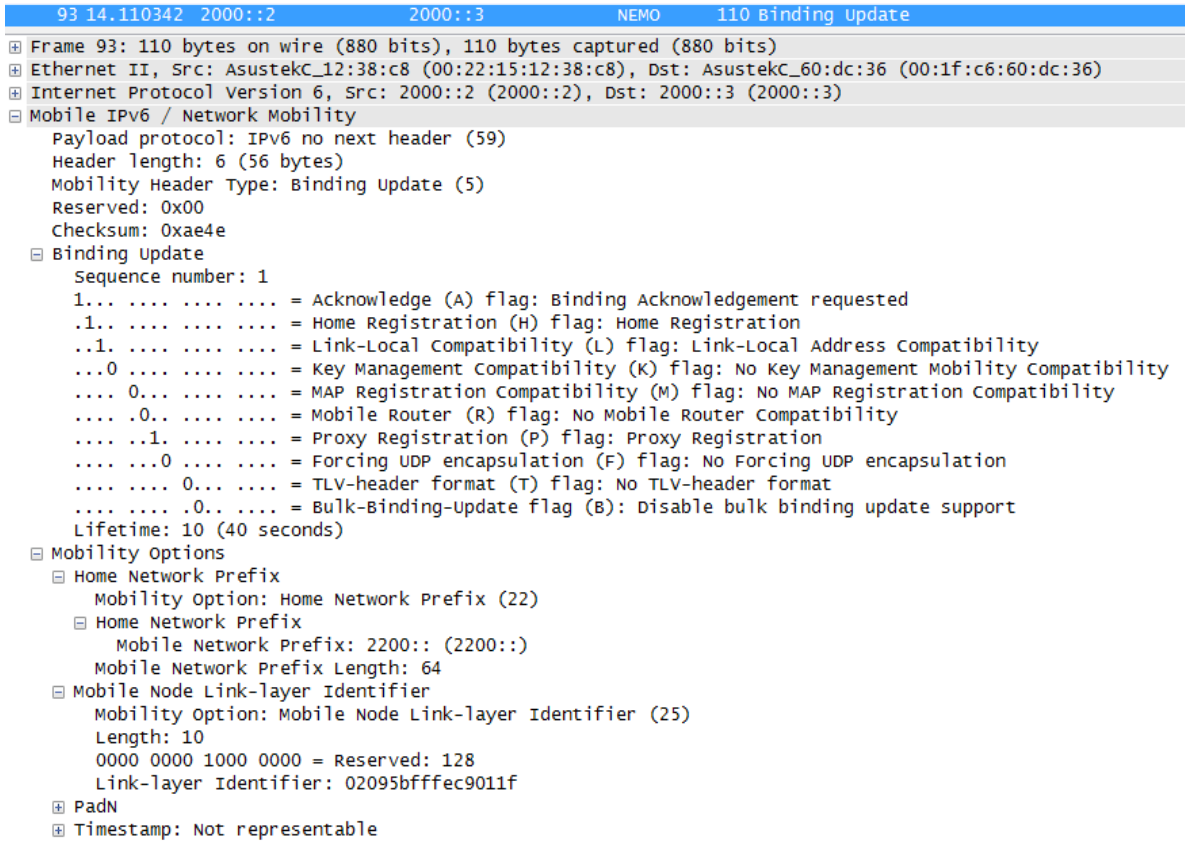


Ilustración 29: Captura de Wireshark de un PBU

- PBA del LMA al MAG:

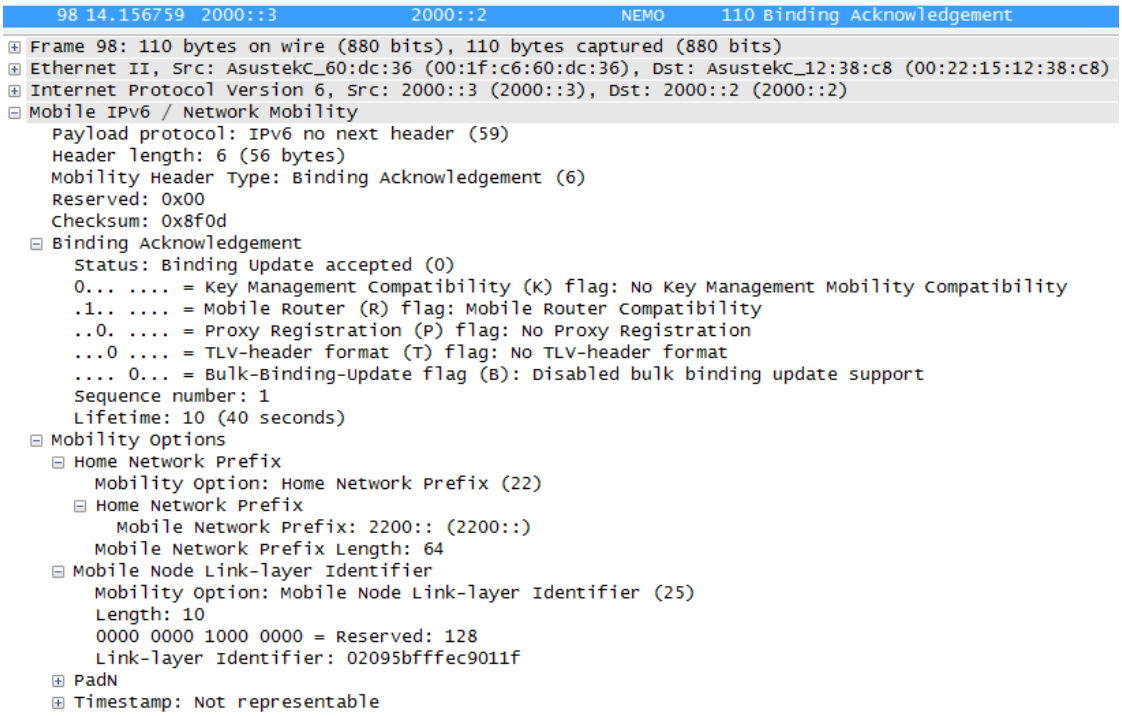


Ilustración 30: Captura de Wireshark de un PBA

- *Router Advertisement* del MAG al MN que incluye el HNP:

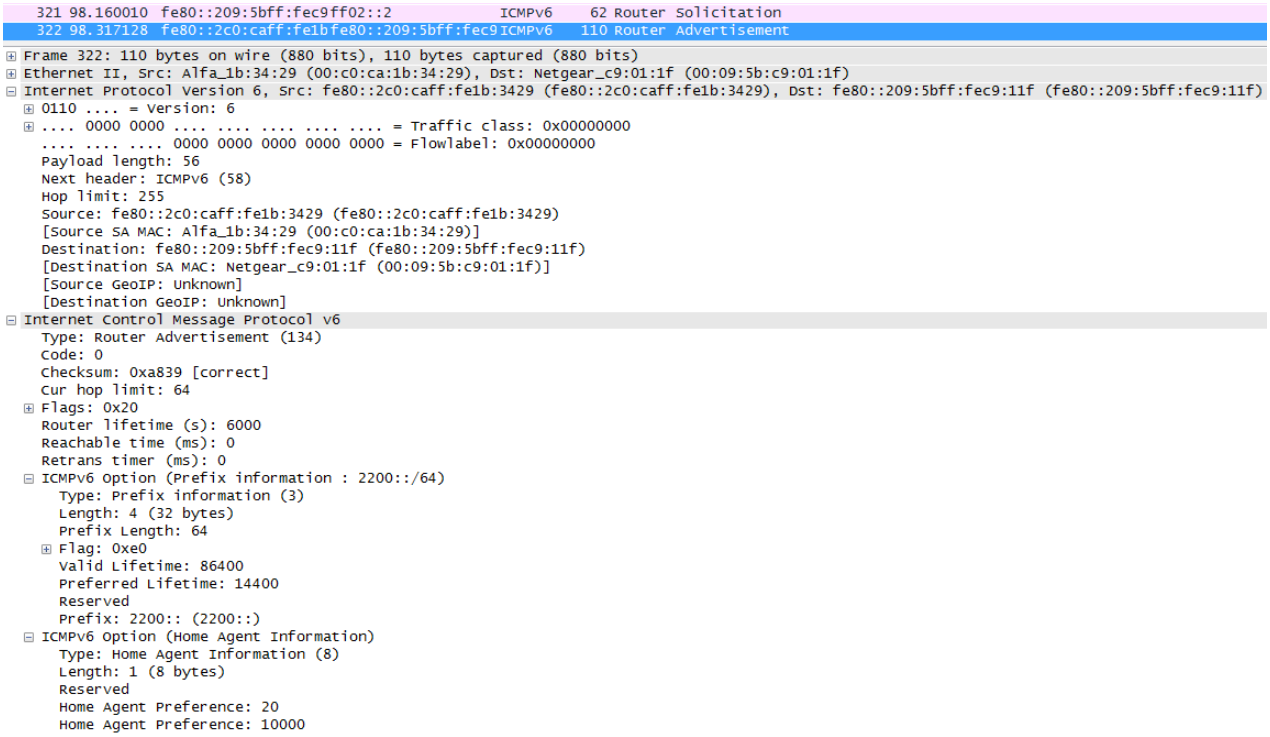


Ilustración 31: Captura de Wireshark de un *Router Advertisement*

Como se puede ver a continuación el MN tiene configurada la dirección IPv6 en la interfaz wlan0 correspondiente al HNP:

```
sgonzalez@caracol:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether d8:50:e6:bc:4b:ae brd ff:ff:ff:ff:ff:ff
    inet 163.117.140.102/24 brd 163.117.140.255 scope global eth0
    inet6 2001:720:410:1020:da50:e6ff:febc:4bae/64 scope global dynamic
       valid_lft 86399sec preferred_lft 14399sec
    inet6 fe80::da50:e6ff:febc:4bae/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:27:19:b0:14:ca brd ff:ff:ff:ff:ff:ff
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:09:5b:c9:01:1f brd ff:ff:ff:ff:ff:ff
    inet6 2200::209:5bff:fec9:11f/64 scope global dynamic
       valid_lft 85677sec preferred_lft 13677sec
    inet6 fe80::209:5bff:fec9:11f/64 scope link
       valid_lft forever preferred_lft forever
```

Ilustración 32: Captura de autoconfiguración de HNP

La implementación de PMIPv6 tiene una herramienta que muestra las entradas de la caché, su utilización es sencilla, basta con usar el siguiente comando en cualquier router que esté corriendo PMIPv6:

telnet localhost 7777

De esta manera podemos comprobar que el MN ha entrado correctamente en el escenario PMIPv6 y a qué MAG se ha conectado:

```
mip6d> pmip
peer addr 0:0:0:0:209:5bff:fec9:11f status PMIP
 Serv_MAG_addr 2000:0:0:0:0:0:0:2 LMA_addr 0:0:0:0:0:0:0:0 local 2000:0:0:0:0:0:0:3
 lifetime 38 / 40 seq 0
mip6d>
```

Ilustración 33: Captura de caché PMIPv6 en el router

Se puede ver como el MN ha entrado correctamente al dominio PMIPv6, su MAG es MAG2 (2000::2) y su LMA (2000::3). También se puede ver que el comando ha sido ejecutado en el LMA.

4.4. Tráfico multicast IPv6 y MCProxy

Una vez comprobado el correcto funcionamiento del escenario PMIPv6 el siguiente paso es conseguir mandar tráfico multicast a través del dominio PMIPv6. Para ello los routers han de ejecutar el protocolo MLDv2 [4], para ello se ha utilizado mcproxy [9]. Existe una versión de este programa compatible con OpenWRT en su última versión Barrier Breaker así que en este caso no ha sido necesario realizar la compilación cruzada como en el caso de PMIPv6. La configuración detallada de mcproxy se muestra en el anexo B.7.2. Configuración mcproxy.

Una vez mcproxy esté correctamente configurado basta con su ejecución en los routers para que el tráfico multicast se redirija por las interfaces indicadas en el archivo de configuración. MCProxy mostrará todos los temporizadores y las suscripciones activas en ese router como se puede ver a continuación:

```
##-- downstream interface: eth0.2 (index:6) --##
querier version: MLDv2
is querier: true
general query timer: 2sec
startup query count: 0
subscribed groups: 6
-- group address: ff02::2
        MLDv2, EXCLUDE_MODE(234sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ff00:0
        MLDv2, EXCLUDE_MODE(234sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ff00:1
        MLDv2, EXCLUDE_MODE(234sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ff00:2
        MLDv2, EXCLUDE_MODE(233sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ff12:38c8
        MLDv2, EXCLUDE_MODE(233sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ff12:391a
        MLDv2, EXCLUDE_MODE(234sec)
        requested list(#0):
        exclude_list(#0):
```

Ilustración 34: Ejemplo de ejecución de mcproxy

Para ejecutar mcproxy se utilizará el siguiente comando en la terminal del router:

mcproxy -dsvv -f mcproxy.conf

Una vez creado el escenario final con la ejecución en segundo plano de PMIPv6 y mcproxy necesitamos evaluar la transmisión y recepción de tráfico multicast, para ello es necesario usar diferentes programas que generen este tipo de tráfico. En nuestro Trabajo Fin de Grado hemos utilizado diferentes programas como MCFirst y MCSender. Su instalación está detallada en el anexo C de esta memoria.

El funcionamiento de MCSender y MCFirst es similar al de un ping, mandando pequeños paquetes con una marca de tiempo de manera continua.

Para su ejecución se utilizarán los siguientes comandos en terminal (se ha puesto como ejemplo envío a través de la interfaz eth1 y recepción a través de wlan0 como en el escenario final):

- MCSender (Envío de tráfico multicast): **sudo mcsender -t3 ff15::1:1234 -ieth1**

- MCFirst (Recepción de MCSender): **mcfirst -6 -I wlan0 ff15::1 1234 -c 1000**

A continuación se puede ver un ejemplo de envío y recepción de tráfico a través de las herramientas MCSender y MCFirst:

```
sgonzalez@sapo:~$ sudo mcsender -t3 ff15::1:1234 -ieth1
[sudo] password for sgonzalez:
```

Ilustración 35: Ejemplo de ejecución de mcsender

```
sgonzalez@caracol:~$ mcfirst -6 -I wlan0 ff15::1 1234 -c 1000
mcfirst joined (*,G) = (*,ff15::1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 31.173 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 32.416 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 33.809 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 792.150 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 1792.300 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 2792.435 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 3792.549 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 4793.333 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 5792.800 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 6792.953 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 7793.045 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 8793.762 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 9793.271 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 10804.149 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 11793.475 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 12794.822 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 13793.723 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 14793.823 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 15794.205 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 16794.655 ms (ttl/hops 1)
```

Ilustración 36: Ejemplo de ejecución de mcfirst

Ahora vamos a ver los paquetes capturados a través de la herramienta wireshark:

Al iniciar mcfirst lanza un Multicast Listener Report Message (Join) a FF15::1:

```
    3 0.004031   fe80::209:5bff:fec9ff02::16          ICMPv6     90 Multicast Listener Report Message v2
⊞ Frame 3: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
⊞ Ethernet II, Src: Netgear_c9:01:1f (00:09:5b:c9:01:1f), Dst: IPv6mcast_16 (33:33:00:00:00:16)
⊞ Internet Protocol Version 6, Src: fe80::209:5bff:fec9:11f (fe80::209:5bff:fec9:11f), Dst: ff02::16 (ff02::16)
⊟ Internet Control Message Protocol v6
    Type: Multicast Listener Report Message v2 (143)
    Code: 0
    Checksum: 0x1206 [correct]
    Reserved: 0000
    Number of Multicast Address Records: 1
  ⊟ Multicast Address Record Changed to exclude: ff15::1
      Record Type: Changed to exclude (4)
      Aux Data Len: 0
      Number of Sources: 0
      Multicast Address: ff15::1 (ff15::1)
```

Ilustración 37: *Multicast Listener Report Message* de mcfirst

Este es el paquete que envía mcsender hacia mcfirst con 46 bytes de datos:

```
    4 0.027722   fe80::227:19ff:feb0ff15::1          UDP      108 Source port: 35755  Destination port: 1234
⊞ Frame 4: 108 bytes on wire (864 bits), 108 bytes captured (864 bits)
⊞ Ethernet II, Src: Alfa_1b:34:29 (00:c0:ca:1b:34:29), Dst: IPv6mcast_01 (33:33:00:00:00:01)
⊞ Internet Protocol Version 6, Src: fe80::227:19ff:feb0:1d8 (fe80::227:19ff:feb0:1d8), Dst: ff15::1 (ff15::1)
⊟ User Datagram Protocol, Src Port: 35755 (35755), Dst Port: 1234 (1234)
    Source Port: 35755 (35755)
    Destination Port: 1234 (1234)
    Length: 54
  ⊞ Checksum: 0xc6c4 [validation disabled]
    [Stream index: 0]
⊟ Data (46 bytes)
    Data: 7468697320697320746865207465737420 6d657373616765...
    [Length: 46]
```

Ilustración 38: Paquete enviado por mcsender

A continuación se muestran los tres escenarios utilizados para probar su funcionamiento:

1) Dos PC directamente conectados:

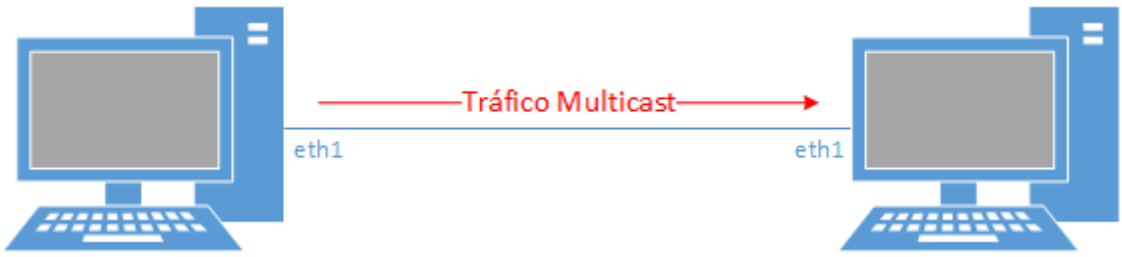


Ilustración 39: Escenario multicast con dos PC

Este escenario es sencillo, punto a punto, sin ningún router intermedio. En este caso mcproxy no será necesario. El objetivo de este escenario es el de familiarizarse con las herramientas de envío y recepción de tráfico multicast para poder utilizarlas en escenarios más complejos.

2) Dos PC con un router entre ellos:

Ilustración 40: Escenario multicast con dos PC y un router intermedio

Una vez familiarizado con las herramientas utilizadas en el escenario anterior se aumentará la dificultad introduciendo un router intermedio que ha de enviar el tráfico de un PC a otro. En este escenario será necesario el uso de mcproxy en el router intermedio para que el tráfico llegue de un nodo/terminal a otro.

3) Dominio PMIPv6, escenario final:

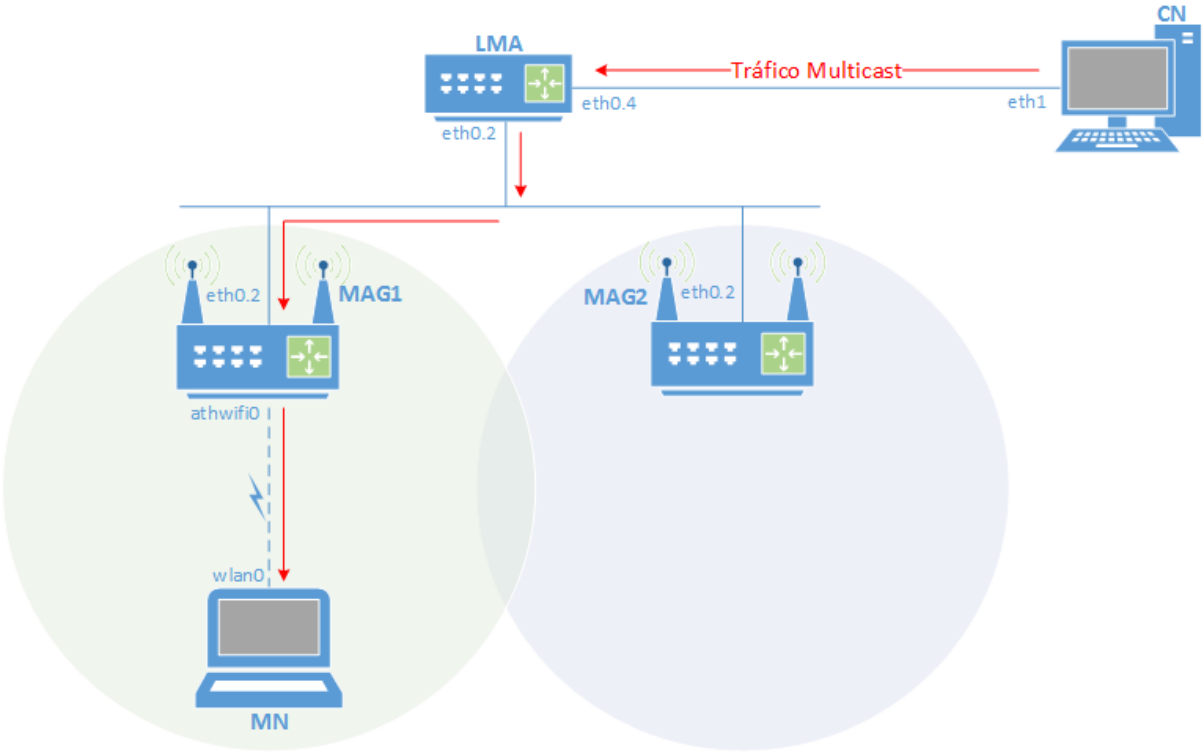


Ilustración 41: Escenario multicast final con PMIPv6

Este es uno de los objetivos de este Trabajo Fin de Grado, poder enviar tráfico multicast a través de un dominio PMIPv6. En este tercer escenario se ejecutarán en segundo plano tanto mcproxy como PMIPv6 para que el tráfico multicast llegue del CN al MN. A partir de ahora todas las pruebas se harán en este escenario.

4.5. Handoff PMIPv6 con multicast

Una vez definido el escenario final de pruebas se va a evaluar el cambio de MAG mientras se realiza una transmisión continua de tráfico multicast, esto nos permitirá medir la calidad del *handoff* en términos de redirección del tráfico.

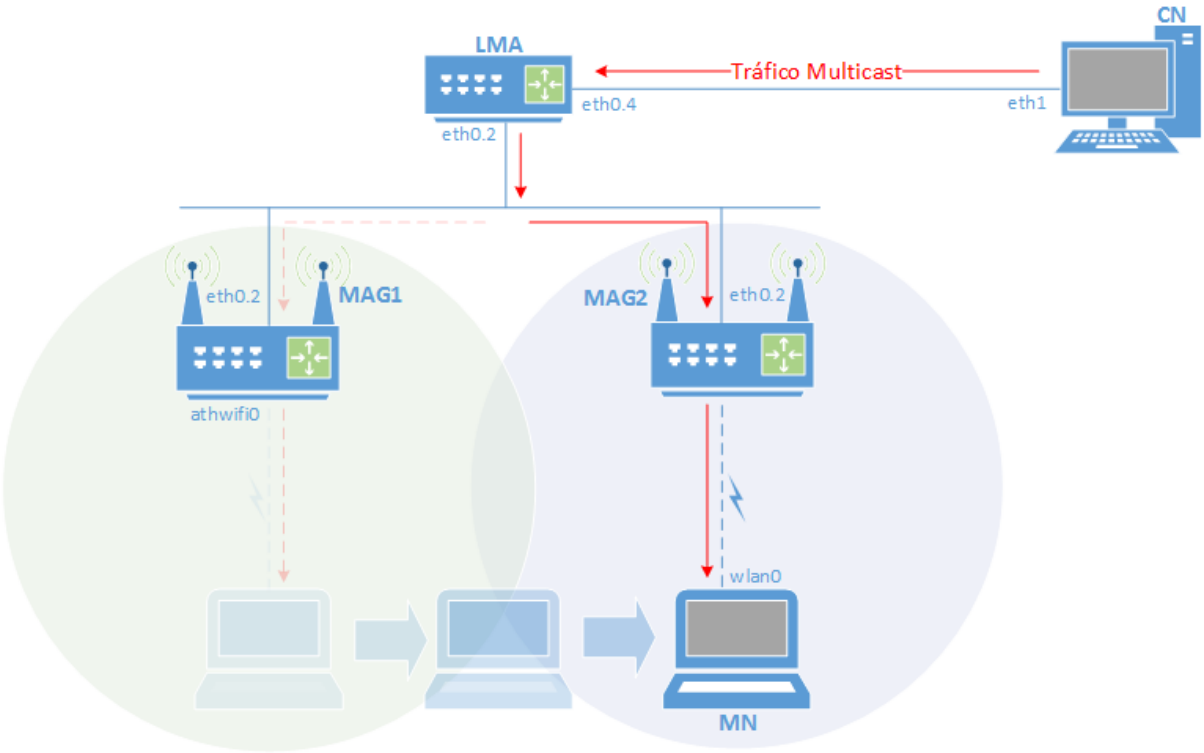


Ilustración 42: *Handoff* en escenario final

Cuando el MN cambia de MAG hay una pérdida de paquetes multicast en el receptor hasta que el tráfico se redirecciona a través del dominio PMIPv6. Esta pérdida de paquetes es causada por el traspaso (a nivel 2 y 3), así como por los temporizadores de MLDv2. Estos temporizadores están definidos según el estándar marcado por la RFC 4604 [4] en el código de mcproxy. Dentro de todos los temporizadores hay tres que son de vital importancia en la rapidez en la redirección del tráfico y son el *Query Interval*, con un valor por defecto de 125 segundos, el *Startup Query Interval*, con un valor por defecto de (*Query Interval*)/4 y el *Query Response Interval*, con un valor por defecto de 10 segundos.

El *Query Interval* y el *Startup Query interval* tienen realmente la misma función, preguntar por suscriptores multicast en la red, el *Query Response Interval* tiene la función de retrasar la respuesta un intervalo aleatorio de tiempo. En la primera ejecución del protocolo MLDv2 se utilizará el *Startup Query Interval*, una vez este se agote entrará en escena el *Query Interval*. El *Startup Query Interval* tiene un valor inferior para que en su primera ejecución pueda obtener la información multicast de la red de manera más rápida. En la

ejecución de mcproxy se puede ver el valor de los *Query Interval* y *Startup Query Interval* y como van decreciendo en el tiempo. Se puede observar en la siguiente imagen:

```
##-- downstream interface: athwifi0 (index:5) --##
querier version: MLDv2
is querier: true
general query timer: 124sec
startup query count: 0
subscribed groups: 3
-- group address: ff02::fb
        MLDv2, EXCLUDE_MODE(138sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff02::1:ffc9:11f
        MLDv2, EXCLUDE_MODE(138sec)
        requested list(#0):
        exclude_list(#0):
-- group address: ff15::1
        MLDv2, EXCLUDE_MODE(178sec)
        requested list(#0):
        exclude_list(#0):
```

Ilustración 43: *Query Interval* en mcproxy

Como se puede ver en la ilustración 43 el temporizador *Query Interval* tiene otro nombre en mcproxy: *general query timer*. Cada vez que este temporizador finaliza se manda un *Query* para preguntar en la red a cerca de las suscripciones multicast.

Hemos observado que en nuestra implementación de PMIPv6 se produce un comportamiento que no sigue fielmente la filosofía de la especificación. Cada vez que un nodo móvil entra en un dominio PMIPv6 debe crearse un nuevo enlace punto a punto (o interfaz lógica) entre el MAG y cada MN, en nuestra implementación esto no sucede, ya que existe una única interfaz. En MLDv2 cada vez que aparece una nueva interfaz se envía un Query para actualizar las suscripciones, por lo que el tiempo de reanudación del tráfico en el *handoff* queda definido por el Query Response Interval que retrasa la respuesta del MN.

En nuestra implementación de PMIPv6 el *handoff* no queda definido únicamente por el *Query Response Interval*, a este hay que añadir el valor de los *Query Interval/General Query*. El valor de este temporizador puede llegar a ser muy alto (125 segundos), y en un *handoff* el MN puede quedarse hasta 125 segundos sin recibir el tráfico multicast que estaba recibiendo en el antiguo MAG, hasta que se redirige el tráfico. Vamos a ilustrar este suceso con un ejemplo real producido en nuestro escenario:

```
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 63738.439 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 64738.511 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 65738.638 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 66738.771 ms (ttl/hops 1)
```

```
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 161942.565 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 162751.401 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 163751.681 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 164751.762 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 165751.869 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 166751.997 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 167751.970 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 168753.237 ms (ttl/hops 1)
Received 46 bytes from fe80::227:19ff:feb0:1d8%wlan0 after 169752.448 ms (ttl/hops 1)
```

Ilustración 44: *Handoff* PMIPv6 con tráfico multicast con temporizadores por defecto

En este ejemplo se puede ver con claridad como el receptor del tráfico (MN) se queda sin recibir tráfico desde el instante 66.738,771 ms hasta el instante 161.942,565 ms, lo que se traduce en un periodo de tiempo aproximado de 95 segundos hasta la correcta redirección del flujo multicast.

La solución estándar definida en la RFC de MLDv2 [4] define unos valores tan altos de los temporizadores para no saturar la red con el envío de *Queries* periódicos, intentado distanciarlos en el tiempo. Como se puede ver no es una solución eficiente en escenarios de movilidad, ya que cuando el nodo móvil cambie su punto de acceso puede tener grandes pérdidas de tráfico.

Este Trabajo Fin de Grado tiene como objetivo evaluar este cambio de punto de acceso, para ello se realizará un número alto de medidas y se crearán gráficas que muestren los resultados. Como objetivo final también se buscará una manera de mejorar estos tiempos y se realizará una comparación con las medidas tomadas anteriormente.

4.6. Mejoras del tiempo de traspaso de PMIPv6 con tráfico multicast

La poca eficiencia que hemos detectado en la transmisión de datos multicast en el *handoff* en PMIPv6 nos ha llevado a buscar una solución. Esta solución está definida en la RFC 6636 [11] y está basada en la modificación de los temporizadores de MLDv2 [4] para agilizar la redirección del tráfico multicast. Se puede consultar toda la información relacionada con los temporizadores en la sección 3.4.2.

Los temporizadores que se van a modificar para mejorar el rendimiento de multicast en una red inalámbrica son:

1) *Query Interval*: es el temporizador entre *Queries*, por defecto vale 125 segundos. Se puede incrementar su valor a 150 segundos en redes con un gran número de hosts para minimizar el número de reportes y disminuir el consumo de batería en dispositivos móviles, este caso no muestra nuestra situación. También se puede disminuir a 60-90 segundos para agilizar la sincronización de las suscripciones, esto aumentará el consumo de batería en los dispositivos móviles. En este Trabajo Fin de Grado se establecerá el *Query Interval* en 60 segundos.

2) *Query Response Interval*: su valor por defecto es 10 segundos. Se puede aumentar hasta 20 segundos para reducir la congestión en los enlaces pero aumentará la latencia. Se puede disminuir a 5 segundos para agilizar el descubrimiento de hosts y la sincronización. En este Trabajo Fin de Grado se establecerá el *Query Response Interval* en 5 segundos.

3) *Last Member Query Timer* (LMQT) y *Last Listener Query Timer* (LLQT): su valor por defecto es 1 segundo. se puede aumentar a 2 segundos en enlaces con muchas pérdidas aumentando la latencia o se puede mantener su valor por defecto. En este caso mantendremos el valor por defecto de 1 segundo.

4) *Startup Query Interval*: es el *Query interval* al inicio de la ejecución, su valor por defecto es ¼ del *Query Interval*. Se puede disminuir su valor a 1 segundo para acortar la actualización de los suscriptores al inicio de la ejecución. En este Trabajo Fin de Grado se va a decrementar su valor a 1 segundo.

5) *Robustness Variable*: su valor por defecto es 2. Se puede mantener el valor original o decrementarlo a 1 para reducir las retransmisiones. En este Trabajo Fin de Grado se va a mantener el valor por defecto de 2.

Todos estos temporizadores están definidos en el programa mcproxy, para modificarlos es necesario acceder al código fuente de la versión del programa desarrollada para OpenWRT y modificar estos valores. Una vez estén modificados se compilará únicamente el paquete de mcproxy para reinstalarlo en los routers. En el anexo B.4. Instalación y compilación de paquetes se explica la manera de realizar la compilación cruzada de los paquetes de mcproxy y su instalación en el router.

Una vez analizado el código original de mcproxy se determinó que la definición de los temporizadores se encontraba en el archivo:

timers_values.hpp

Situado en el directorio:

~/openwrt/trunk/build_dir/target-mipsel_mips32_uClibc-0.9.33.2/mcproxy-2014-05-02-bbb2e7ee230c172e68766946e4b4e48f7449ee15/mcproxy/include/proxy/

Aquí se puede ver la porción de código donde están definidos los temporizadores de MLDv2 en mcproxy:

```
struct timers_values_tank {
    unsigned int robustness_variable = 2;
    std::chrono::seconds query_interval = std::chrono::seconds(125);
    std::chrono::milliseconds query_response_interval = std::chrono::milliseconds(10000); //Max Response Time/Delay
    std::chrono::seconds startup_query_interval = query_interval / 4;
    unsigned int startup_query_count = robustness_variable;
    std::chrono::milliseconds last_listener_query_interval = std::chrono::milliseconds(1000);
    unsigned int last_listener_query_count = robustness_variable;
    std::chrono::milliseconds unsolicited_report_interval = std::chrono::milliseconds(1000);
};
```

Ilustración 45: Código de los temporizadores de mcproxy

Se procederá a la modificación de los valores de los temporizadores de manera acorde a la definida en la RFC 6636, a continuación se evaluará la mejora obtenida en nuestro escenario PMIPv6 respecto a los valores por defecto.

```
struct timers_values_tank {
    unsigned int robustness_variable = 2;
    std::chrono::seconds query_interval = std::chrono::seconds(60);
    std::chrono::milliseconds query_response_interval = std::chrono::milliseconds(5000); //Max Response Time/Delay
    std::chrono::seconds startup_query_interval = std::chrono::seconds(1);
    unsigned int startup_query_count = robustness_variable;
    std::chrono::milliseconds last_listener_query_interval = std::chrono::milliseconds(1000);
    unsigned int last_listener_query_count = robustness_variable;
    std::chrono::milliseconds unsolicited_report_interval = std::chrono::milliseconds(1000);
};
```

Ilustración 46: Código de los temporizadores modificados de mcproxy

5. Resultados

En este capítulo se mostrarán los resultados de las pruebas realizadas y la evaluación de estos resultados.

La pruebas se han llevado a cabo en el escenario definido en las ilustraciones 38 y 39.

Se han efectuado 30 cambios de MAG con un envío continuo de tráfico multicast entre el CN y el MN, para evaluar el rendimiento se ha medido el tiempo que tarda el tráfico en volver a llegar al receptor. Se han realizado estas medidas de manera completamente aleatoria, con los temporizadores por defecto de MLDv2 [4] y con los detallados en la RFC 6636 [11].

Para mostrar correctamente la mejora obtenida se mostrarán diferentes gráficas realizadas con las herramientas Highcharts y LibreOffice Calc.

En la siguiente gráfica podemos ver los tiempos obtenidos con temporizadores por defecto de MLDv2 en orden cronológico:

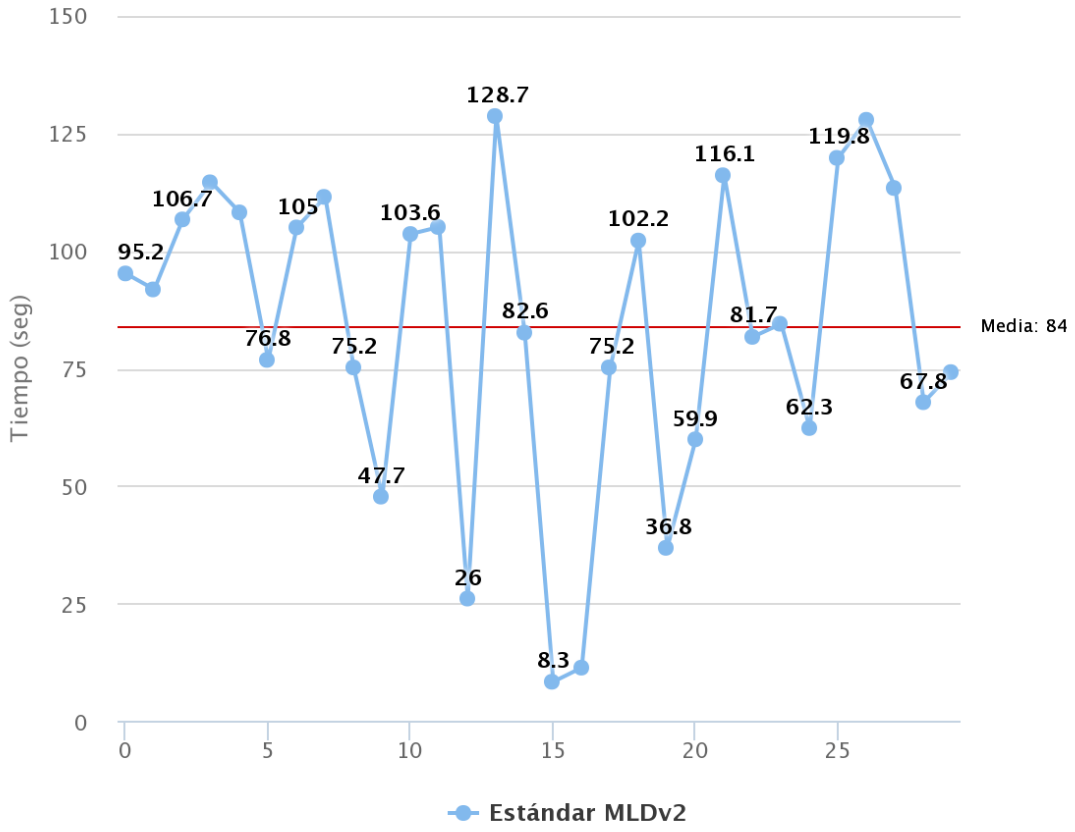


Ilustración 47: Tiempo de reanudación de tráfico multicast con temporizadores por defecto de MLDv2

A continuación se va a mostrar la CDF (*Cumulative Distribution Function*) o Función de Distribución Acumulada (FDA) de las muestras obtenidas con los temporizadores por defecto de MLDv2:

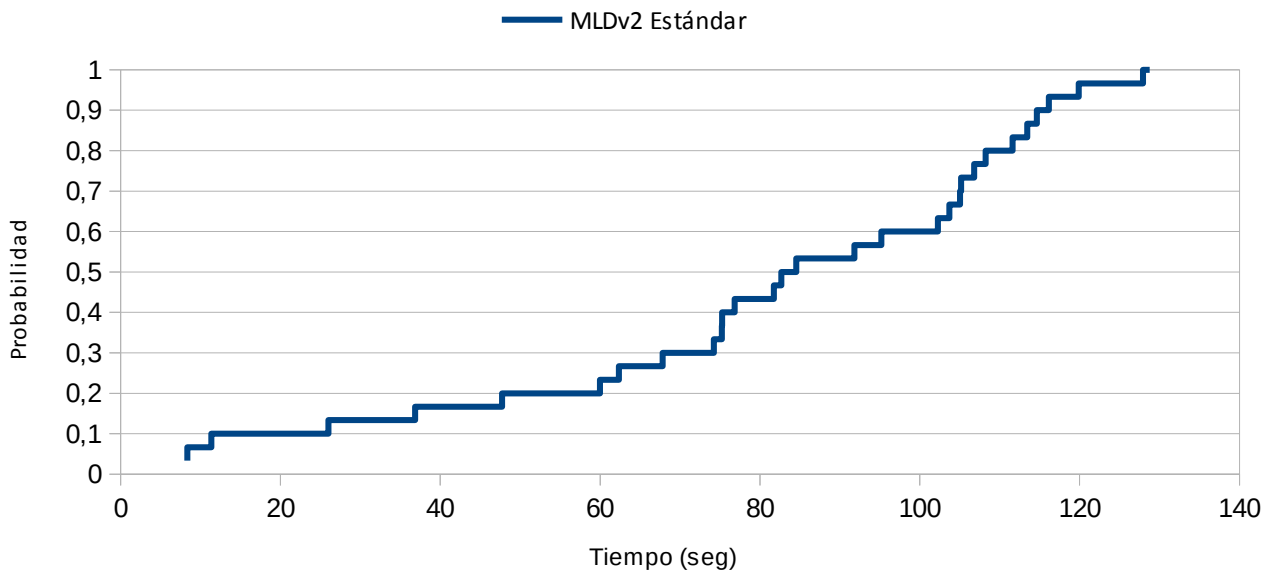


Ilustración 48: Función de Distribución Acumulada con los temporizadores por defecto de MLDv2

En las ilustraciones 44 y 45 se puede ver que las medidas realizadas oscilan entre un valor máximo de 128.7 segundos y un valor mínimo de 8.3 segundos. Este abanico de tiempo se debe a que el *Query Interval* tiene un valor por defecto de 125 segundos. El valor medio de estas mediciones es de 84 segundos y con una desviación estándar de 32.8 segundos.

A continuación vamos a mostrar los tiempos obtenidos con los temporizadores definidos en la RFC 6636 en orden cronológico:

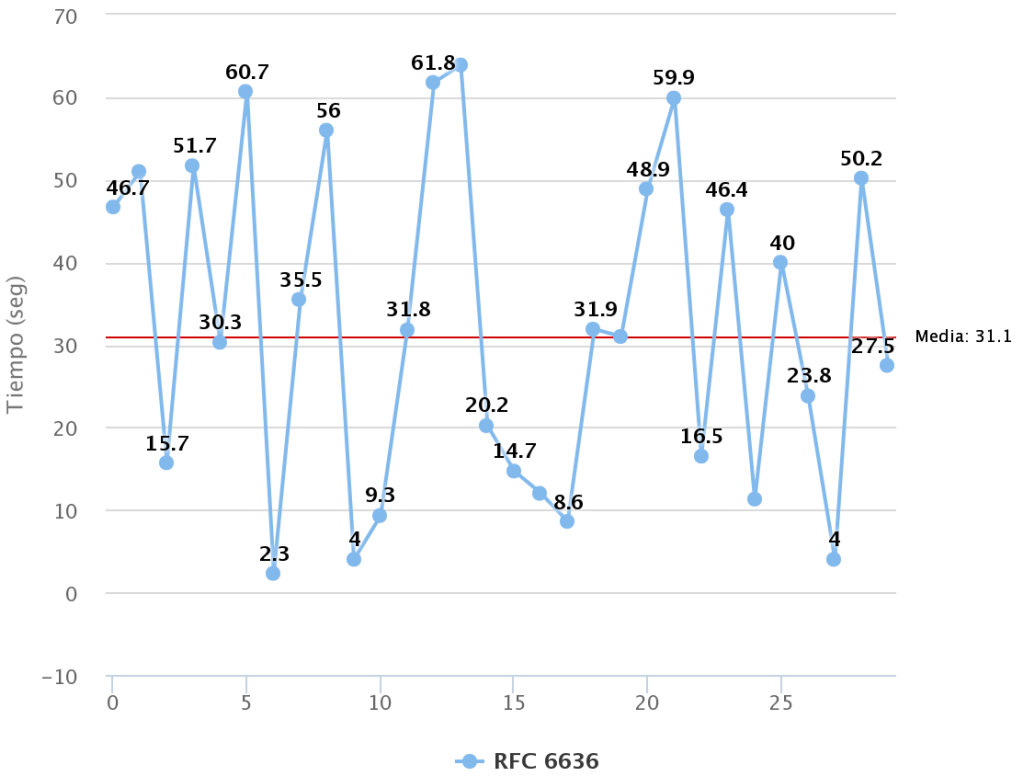


Ilustración 49: Tiempo de reanudación de tráfico multicast con temporizadores definidos en RFC6636

En la siguiente gráfica se va a mostrar la CDF (*Cumulative Distribution Function*) o Función de Distribución Acumulada (FDA) de las muestras obtenidas con los temporizadores de la RFC6636:

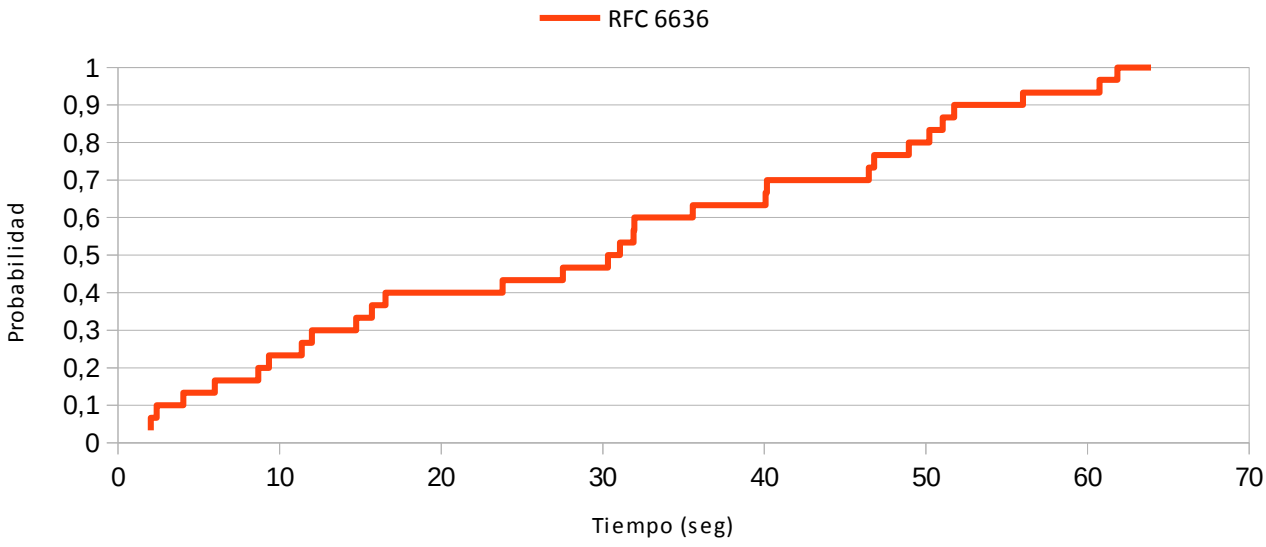


Ilustración 50: Función de Distribución Acumulada con los temporizadores definidos en RFC6636

En las ilustraciones 46 y 47 se muestran las medidas de la segunda prueba, estos valores oscilan entre un valor máximo de 63.9 segundos y un valor mínimo de 2 segundos. La media de todas las muestras es de 31.1 segundos y la desviación estándar es de 19.7 segundos.

Los resultados obtenidos no son los esperados debido a la implementación de PMIPv6 que hemos utilizado. Los valores ideales deberían ser inferiores, en el cambio de MAG se debería crear una interfaz nueva que desataría automáticamente el envío de un *Query* para actualizar las suscripciones, por lo que el tiempo de reanudación del tráfico debería ajustarse al *Query Response Interval*.

Aún así cabe destacar la mejora obtenida con la implementación de la RFC6636 en nuestro escenario, el tiempo de reanudación ha descendido una media de 52.9 segundos con el decremento del valor de los temporizadores.

Para ver de manera más clara la mejora obtenida vamos a introducir dos gráficas, una muestre cronológicamente los valores de las dos pruebas y la gráfica de las dos CDF:

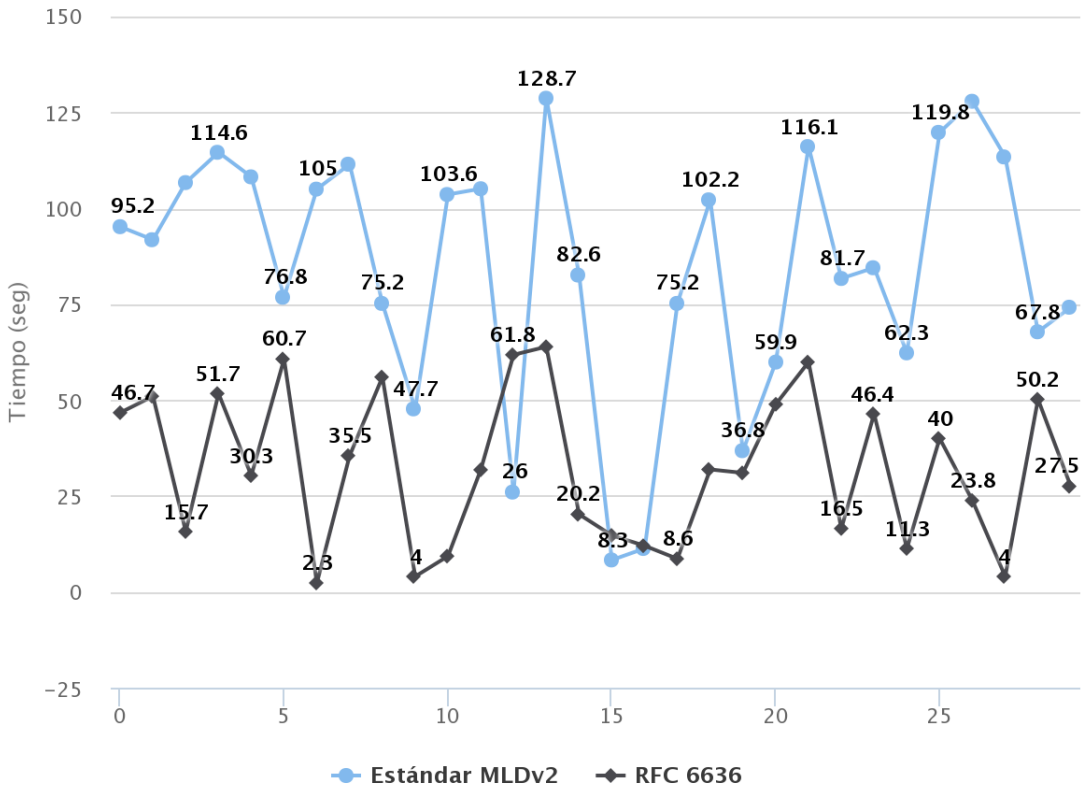


Ilustración 51: Tiempo de reanudación de tráfico multicast de las dos pruebas juntas

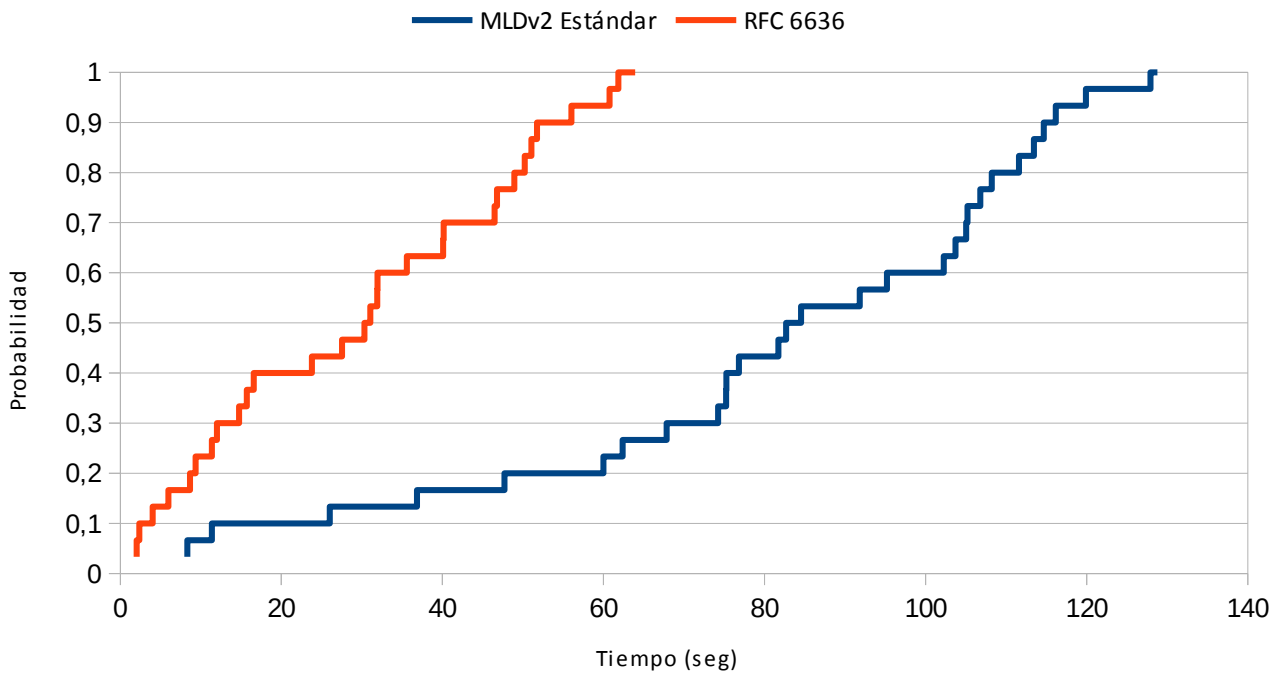


Ilustración 52: Función de Distribución Acumulada de las dos pruebas juntas

La disminución en el tiempo de reanudación del tráfico es importante pero conlleva una serie de desventajas:

- El número de *Queries* periódicos se duplica al disminuir los temporizadores, saturando más la red.
- Se produce un aumento en la carga de la CPU del router debido a que debe procesar más paquetes.
- Se produce un aumento del consumo energético al aumentar la actividad de los routers.

6. Conclusiones y futuros trabajos

En este capítulo se exponen las conclusiones obtenidas al realizar este trabajo fin de grado, así como los posibles trabajos futuros que se podrían realizar para continuar este proyecto.

6.1. Conclusiones

Este Trabajo Fin de Grado se centra en la evaluación de los mecanismos de soporte de tráfico multicast con movilidad basada en red. Para ello se ha construido un escenario PMIPv6, se le ha añadido funcionalidad MLDv2 a los routers y se han realizado pruebas de envío y recepción de tráfico multicast para evaluar su rendimiento.

Una vez realizadas estas pruebas y analizados los resultados se ha llegado varias conclusiones:

1. La definición estándar de MLDv2 no tiene un rendimiento óptimo en escenarios de movilidad.
2. Se ha observado que el comportamiento de nuestra implementación de PMIPv6 no sigue fielmente la filosofía de la especificación. Esta irregularidad afecta a la creación de enlaces punto a punto entre el MAG y el MN en el dominio PMIPv6.
3. Se ha observado que el tiempo de reanudación del tráfico en nuestro escenario con los temporizadores por defecto de MLDv2 puede ser incluso superior a dos minutos, con un valor medio de 84 segundos.

Para intentar reducir este tiempo y mejorar la experiencia en el nodo móvil se ha implementado la RFC6636. Este documento propone una modificación de los temporizadores de MLDv2, esta modificación busca reducir el tiempo de reanudación del tráfico sin congestionar la red. Una vez implementada la mejora se han realizado las mismas pruebas de evaluación de rendimiento.

Terminadas las pruebas con las dos implementaciones se han comparado los resultados. Se ha llegado a la conclusión de que la implementación de la RFC6636 produce una mejora notable en nuestro escenario, reduciendo el tiempo de reanudación del tráfico en una media de 53 segundos. Aún así el tiempo medio de reanudación tras la implementación de la RFC6636 es de 31.1 segundos, un valor todavía elevado para una experiencia de usuario óptima.

Se han buscado alternativas que mejoren la eficiencia de la RFC6636 y se han incluido en el siguiente apartado: 6.2.

6.2. Futuros Trabajos

Se van a enumerar diferentes propuestas como futuras mejoras en base a este proyecto:

- **Uso de otra implementación PMIPv6:** se ha observado que el comportamiento de nuestra implementación de PMIPv6 no sigue fielmente la filosofía de la especificación. En el futuro se podría analizar la mejora producida por la RFC6636 en una implementación de PMIPv6 sin esta irregularidad.
- **Evaluación del traspaso de manera cualitativa:** la evaluación de resultados en este proyecto se ha realizado de manera cuantitativa con las herramientas mcfirst y mcsender. En un futuro se puede realizar esta evaluación de manera cualitativa con un streaming multicast en tiempo real, para medir el impacto en un uso real.
- **Ampliación del escenario:** se puede ampliar el escenario con más LMAs y MAGs para realizar un análisis más complejo.
- **Implementación de la RFC 7161:** como solución o mejora al *handoff* PMIPv6 con tráfico multicast se puede implementar la RFC 7161 [29]. Este documento propone un nuevo formato para los PBU y PBA para añadir información multicast en la cabecera. Para realizar esta mejora debería modificarse el código de mcproxy y el de la implementación PMIPv6 para que sea posible el intercambio de información.
- **Implementación de la RFC 7028:** para mejorar el soporte multicast en un dominio PMIPv6 se puede implementar las optimizaciones propuestas en la RFC 7028 [30] basadas en Multicast Tree Mobility Anchor (MTMA) y Direct Routing.

7. Conclusions and future works

In this chapter the conclusions obtained doing this Bachelor Thesis are exposed, as well as the possible future works that could be done to continue this project.

7.1. Conclusions

This Final Bachelor Thesis focuses on the evaluation of the mechanisms of multicast traffic support with network-based mobility. For that we have built a PMIPv6 scenario with MLDv2 functionality, in this scenario we have made multicast sending and receiving tests for evaluate the performance.

Once we finished the tests and the analysis we can conclude:

1. The standard implementation of MLDv2 does not have an optimal performance in mobility scenarios.
2. We have detected that the behaviour of our PMIPv6 implementation doesn't follow faithfully the philosophy of the specification. This irregularity is related to the creation of point-to-point links between the MAG and the MN in the PMIPv6 domain.
3. We have shown that the time of resumption of the traffic reception is too high, in several times it has reached a value of two minutes, with a mean value of 84 seconds.

To reduce the time and improve the experience in the mobile node we implement the RFC6636. This document proposes a MLDv2 timer modification to reduce the time of resumption without congestion in the network. Once implemented we have made the same test for measure the performance.

Finished the testing of the two implementations we compared the results. We can conclude: the RFC6636 implementation produces a big improvement in our scenario, reducing the resumption time an average of 53 seconds. Even so the average of the resume time after the RFC6636 implementation is 31.1 seconds, still a high value for optimal user experience.

We have sought different alternatives to improve the efficiency of the RFC6636, they are included in the next section 7.2.

7.2. Future Works

Different proposals and future improvements based on this project will be listed in the following lines:

- **Using a different PMIPv6 implementation:** we have detected that the behaviour of our PMIPv6 implementation doesn't follow faithfully the philosophy of the specification. In the future the improvement of RFC6636 could be analysed with a PMIPv6 implementation with a correct behaviour.
- **Qualitative evaluation:** in this project we have evaluated the results quantitatively with mcsender and mcfirst. In the future this evaluation could be qualitative with a multicast live video streaming, to measure the impact in a real use.
- **Expanding the scenario:** the final scenario used in this work could be expanded adding LMAs and MAGs to perform a more complex evaluation.
- **RFC 7161 implementation:** for improve the PMIPv6 handoff with multicast traffic RFC 7161 [29] could be implemented. This document purpose a new format for PBU and PBA to add multicast information in the header. To perform this improvement the mcproxy and PMIPv6 code must be modified to enable the information exchange.
- **RFC 7028 implementation:** to improve multicast support in a PMIPv6 domain the optimizations described in RFC 7028 [30] could be implemented. This optimizations are based in Multicast Tree Mobility Anchor (MTMA) and Direct Routing.

8. Anexos

En este capítulo se detalla la planificación del proyecto, los recursos utilizados y el presupuesto junto a información adicional a cerca de OpenWRT y configuración de programas utilizados en este Trabajo Fin de Grado.

A. Planificación de tareas, recursos y presupuesto

A.1. Planificación de tareas

En este apartado del anexo A se detallará toda la planificación de tareas de este Trabajo Fin de Grado.

Todo el proyecto se ha dividido en 8 grandes tareas:

A. Documentación y estado del arte

B. Instalación y configuración de los routers

C. Instalación de PMIPv6 en el router

D. Instalación de MCProxy en los routers

E. Funcionamiento conjunto de PMIPv6 y MCProxy en los routers

F. Testeo y prueba de rendimiento de multicast en movilidad

G. RFC 6636 estudio, implementación y evaluación

H. Memoria

En la siguiente tabla se divide cada tarea en subtareas y se detalla la duración, los recursos utilizados y las horas totales empleadas. Se ha estimado que los días laborables de cada mes ascienden a 20 días y equivale a 160 horas.

Tarea	Duración (Semanas)	Recursos (ing/mes)	Horas totales
A. Documentación y estado del arte	**7**	**0,4375**	**70**
A.1. Estudio de IPv6	2	0,125	20
A.2. Estudio de PMIPv6	3	0,1875	30
A.3. Estudio de MLDv2	2	0,125	20
B. Instalación y configuración de los routers	**9**	**1,25**	**200**
B.1 Documentación acerca de OpenWRT	1	0,0625	10
B.2. Toma de contacto con Linksys WRT54GL y OpenWRT	1	0,125	20
B.3. Instalación de imagen genérica	1	0,125	20
B.4. Configuración de los routers	1	0,125	20
B.5. Creación de imagen a medida	3	0,5625	90
B.6. Instalación y configuración de imagen a medida	2	0,25	40
C. Instalación de PMIPv6 en el router	**7**	**0,75**	**120**
C.1. Estudio del funcionamiento de la aplicación PMIPv6	1	0,0625	10
C.2. Estudio de los métodos de compilación cruzada	1	0,0625	10
C.3. Compilación cruzada e instalación de PMIPv6 en routers	2	0,375	60
C.4. Testeo y prueba de PMIPv6 en routers	1	0,125	20
C.5. Diseño y creación de escenario final PMIPv6	2	0,125	20
D. Instalación de MCProxy en los routers	**5**	**0,75**	**120**
D.1. Estudio del funcionamiento de MCProxy	1	0,0625	10
D.2. Instalación de MCProxy en los routers	2	0,375	60
D.3. Instalación y prueba de herramientas multicast	1	0,125	20
D.4. Testeo y prueba de MCProxy en los routers	1	0,1875	30
E. Funcionamiento conjunto de PMIPv6 y MCProxy en los routers	**6**	**1**	**160**
E.1. Instalación conjunta en Linksys WRT54GL	2	0,375	60
E.2. Creación de imagen a medida en Asus WL-500GP	1	0,1875	30
E.3. Instalación conjunta en Asus WL-500GP	1	0,1875	30
E.4. Creación del escenario final	2	0,25	40
F. Testeo y prueba de rendimiento de multicast en movilidad	**2**	**0,375**	**60**
F.1. Pruebas con herramientas multicast en escenario final	1	0,1875	30

F.2. Evaluación de los resultados de las pruebas	1	0,1875	30
G. RFC 6636 estudio, implementación y evaluación	**5**	**0,625**	**100**
G.1. Estudio de soluciones para mejorar *handoff* multicast	2	0,125	20
G.2. Implementación RFC 6636 en el escenario final	1	0,1875	30
G.3. Pruebas con herramientas multicast con la RFC 6636	1	0,1875	30
G.4. Evaluación de los resultados de las pruebas	1	0,125	20
H. Memoria	**9**	**1,6875**	**270**
H.1. Redacción de la memoria	8	1,5	240
H.2. Preparación de la presentación	1	0,1875	30
RESULTADO	**50**	**6,875**	**1.100**

Tabla 2: Planificación de Tareas

A continuación se mostrarán dos diagramas de Gantt. El primer diagrama, más general, sólo mostrará las tareas principales. El segundo diagrama, más específico, mostrará las tareas principales junto a las subtareas.

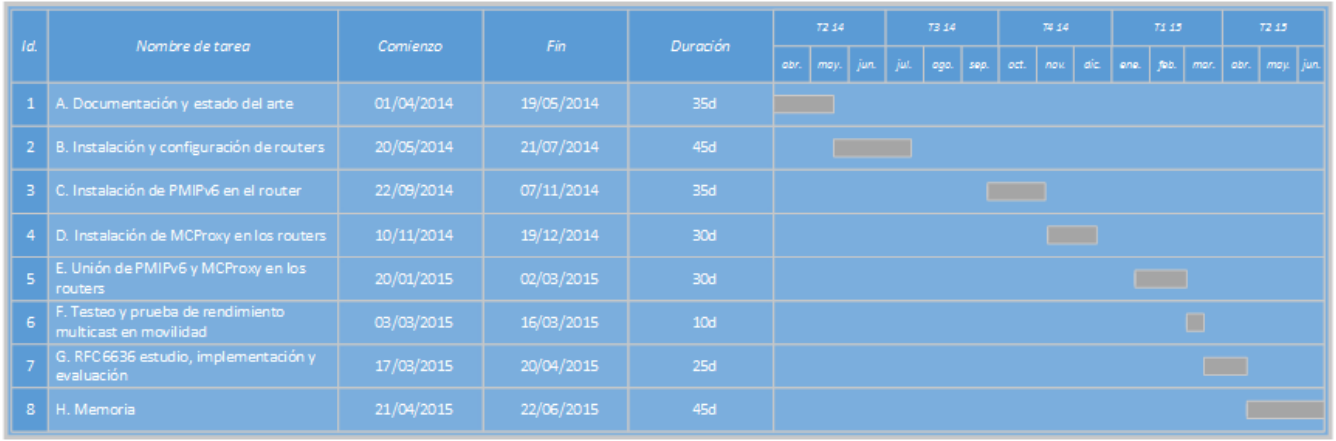
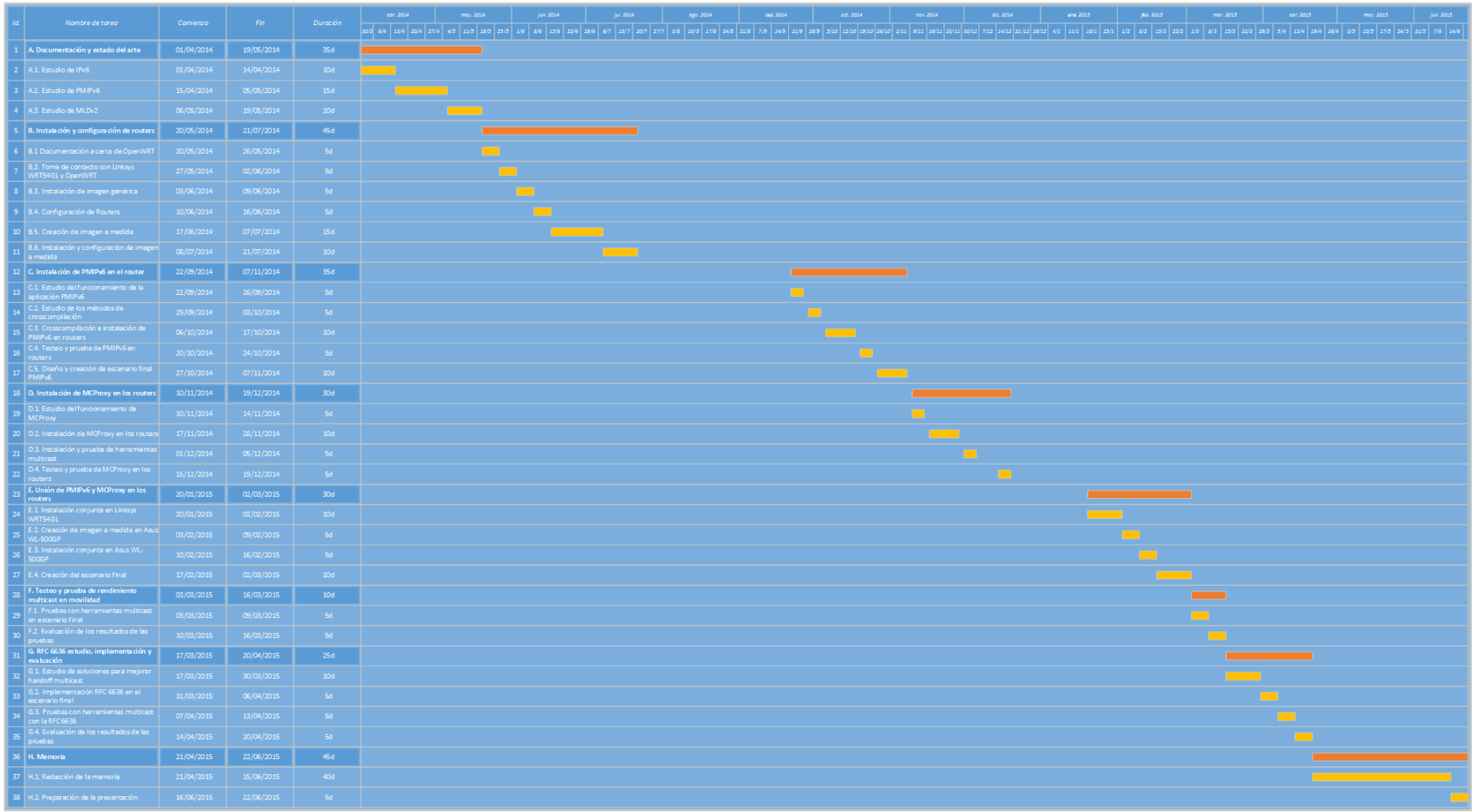


Ilustración 53: Diagrama de Gantt general



Id.	Nombre de tarea	Comienzo	Fin	Duración
1	**A. Documentación y estado del arte**	01/04/2014	19/05/2014	35d
2	A.1. Estudio de IPv6	01/04/2014	14/04/2014	10d
3	A.2. Estudio de PMIPv6	15/04/2014	05/05/2014	15d
4	A.3. Estudio de MLDv2	06/05/2014	19/05/2014	10d
5	**B. Instalación y configuración de routers**	20/05/2014	21/07/2014	45d
6	B.1 Documentación acerca de OpenWRT	20/05/2014	26/05/2014	5d
7	B.2. Toma de contacto con Linksys WRT54GL y OpenWRT	27/05/2014	02/06/2014	5d
8	B.3. Instalación de imagen genérica	03/06/2014	09/06/2014	5d
9	B.4. Configuración de Routers	10/06/2014	16/06/2014	5d
10	B.5. Creación de imagen a medida	17/06/2014	07/07/2014	15d
11	B.6. Instalación y configuración de imagen a medida	08/07/2014	21/07/2014	10d
12	**C. Instalación de PMIPv6 en el router**	22/09/2014	07/11/2014	35d
13	C.1. Estudio del funcionamiento de la aplicación PMIPv6	22/09/2014	26/09/2014	5d
14	C.2. Estudio de los métodos de crosscompilación	29/09/2014	03/10/2014	5d
15	C.3. Crosscompilación e instalación de PMIPv6 en routers	06/10/2014	17/10/2014	10d
16	C.4. Testeo y prueba de PMIPv6 en routers	20/10/2014	24/10/2014	5d
17	C.5. Diseño y creación de escenario final PMIPv6	27/10/2014	07/11/2014	10d
18	**D. Instalación de MCProxy en los routers**	10/11/2014	19/12/2014	30d
19	D.1. Estudio del funcionamiento de MCProxy	10/11/2014	14/11/2014	5d
20	D.2. Instalación de MCProxy en los routers	17/11/2014	28/11/2014	10d
21	D.3. Instalación y prueba de herramientas multicast	01/12/2014	05/12/2014	5d
22	D.4. Testeo y prueba de MCProxy en los routers	15/12/2014	19/12/2014	5d
23	**E. Unión de PMIPv6 y MCProxy en los routers**	20/01/2015	02/03/2015	30d
24	E.1. Instalación conjunta en Linksys WRT54GL	20/01/2015	02/02/2015	10d
25	E.2. Creación de imagen a medida en Asus WL-500GP	03/02/2015	09/02/2015	5d
26	E.3. Instalación conjunta en Asus WL-500GP	10/02/2015	16/02/2015	5d
27	E.4. Creación del escenario final	17/02/2015	02/03/2015	10d
28	**F. Testeo y prueba de rendimiento multicast en movilidad**	03/03/2015	16/03/2015	10d
29	F.1. Pruebas con herramientas multicast en escenario final	03/03/2015	09/03/2015	5d
30	F.2. Evaluación de los resultados de las pruebas	10/03/2015	16/03/2015	5d
31	**G. RFC 6636 estudio, implementación y evaluación**	17/03/2015	20/04/2015	25d
32	G.1. Estudio de soluciones para mejorar handoff multicast	17/03/2015	30/03/2015	10d
33	G.2. Implementación RFC 6636 en el escenario final	31/03/2015	06/04/2015	5d
34	G.3. Pruebas con herramientas multicast con la RFC6636	07/04/2015	13/04/2015	5d
35	G.4. Evaluación de los resultados de las pruebas	14/04/2015	20/04/2015	5d
36	**H. Memoria**	21/04/2015	22/06/2015	45d
37	H.1. Redacción de la memoria	21/04/2015	15/06/2015	40d
38	H.2. Preparación de la presentación	16/06/2015	22/06/2015	5d

Ilustración 54: Diagrama de Gantt extendido

A.2. Recursos

En este Apartado se detallan los recursos hardware y software utilizados para la realización de este Trabajo Fin de Grado.

A.2.1. Hardware

- 2 PC del Laboratorio 4.1.F04 perteneciente al Departamento de Ingeniería Telemática:
  - Procesador: Intel Core i5 4430 (4 núcleos@3GHz, 4 hilos, Turbo Boost@3.2GHz, Caché 6MB)
  - Memoria RAM: 8GB@1333MHz DDR3
  - Almacenamiento: HDD 500GB
  - Conexión: Tarjeta de red PCI inalámbrica Atheros 802.11bg, tarjeta de red PCI alámbrica con salida ethernet 1Gbit/s
- 3 Routers Linksys WRT54GL:
  - Procesador: Broadcom BCM5352@200MHz
  - Memoria RAM: 16MB
  - Memoria Flash: 4MB
  - Salidas Ethernet: 4x10/100
  - Conectividad Inalámbrica: 802.11b/g
- 3 Routers Asus WL-500G Premium v1:
  - Procesador: Broadcom BCM4704@264Mhz
  - Memoria RAM: 32MB
  - Memoria Flash: 8MB
  - Salidas Ethernet: 4x10/100
  - Conectividad Inalámbrica: 802.11b/g
  - USB: 2x2.0

- Ordenador portátil Macbook Pro modelo de 2012:
  - Procesador: Intel Core i5 3210m (2 núcleos@2.5GHz, 4 hilos, Turbo Boost@3.1GHz, Caché 3MB)
  - Tarjeta Gráfica: Intel HD Graphics 4000
  - Memoria RAM: 8GB@1600MHz DDR3
  - Almacenamiento: SSD 128GB
- Ordenador de sobremesa:
  - Procesador: Intel Core i7 4790K (4 núcleos@4GHz, 8 hilos, Turbo Boost@4.4GHz, Caché 8MB)
  - Tarjeta Gráfica: AMD Radeon R9 290
  - Memoria RAM: 8GB@1600MHz DDR3
  - Almacenamiento: SSD 512GB, HDD 4TB

A.2.2. Software

- Sistemas Operativos:
  - Windows 8.1 Pro
  - OS X 10.10 Yosemite
  - GNU/Linux Debian 7 Wheezy
  - OpenWRT 14.07 Barrier Breaker
- Herramientas:
  - Wireshark
  - Gedit
  - MCSender (SMCRoute)
  - MCFirst (SSMPing)
- Herramientas Web:
  - Google Drive
  - Highcharts
- Herramientas de Ofimática:
  - Microsoft Visio 2013
  - LibreOffice
  - Microsoft Office Professional 2013

A.3. Presupuesto

- **Autor:** Sergio González Díaz
- **Departamento de Ingeniería Telemática**
- **Descripción del Proyecto**
  - **Título:** Evaluación de los mecanismos de soporte de tráfico multicast con movilidad basada en red
  - **Duración:** 12 meses
  - **Tasa de costes indirectos:** 20%
- **Presupuesto Total del Proyecto:**
- **Subcontratación de tareas:** No se especifican
- **Otros costes indirectos:** No se especifican

Concepto	Cantidad	Coste (€)	% Proyecto	Dedicación (meses)	Depreciación	Total (€)
PC de laboratorio	2	600	100	12	60	240
Ordenador portátil	1	1.200	100	12	60	240
Ordenador de sobremesa	1	1.500	100	12	60	300
Linksys WRT54GL	3	50	100	12	60	30
Asus WL-500G Premium	3	70	100	12	60	42
Subtotal						852

Tabla 3: Costes de recursos materiales

El cálculo del Coste imputable de los recursos materiales ha sido calculado con la siguiente fórmula:

$$\frac{a}{b} * c * d$$

Dónde: a = dedicación en meses, b = depreciación, c= coste, d = % proyecto

Herramienta de Software	Coste de la licencia (€)
Windows 8.1 Pro	279
Microsoft Visio 2013	739
Microsoft Office Professional 2013	539
Subtotal	1.557

Tabla 4: Costes de herramientas de Software

Según un estudio [31] realizado por el Colegio Oficial de Ingenieros y la Asociación Española de Ingenieros de Telecomunicaciones, el salario medio de un ingeniero de telecomunicaciones es de 49.206€ anuales, para ingenieros con menos de 5 años de experiencia es de 25.730€ anuales. Además del coste deducido del estudiante que ha realizado este Trabajo Fin de Grado, se ha añadido el coste del tutor y los dos directores. Tomando estos valores como referencia obtenemos:

Concepto	Cantidad	Dedicación (ing/mes)	Salario Anual (€)	Total (€)
Ingeniero Senior	3	0,5	49.206	6.150,75
Graduado Ing. Telemática	1	6,875	25.730	14.741,14
Subtotal				19.891,89

Tabla 5: Costes de recursos humanos

Concepto	Total (€)
Recursos materiales	852
Herramientas de Software	1.557
Recursos Humanos	19.891,89
TOTAL	**22.300,89**

Tabla 6: Costes Totales

A.4. Marco Regulador

Todos los protocolos y estándares utilizados en este Trabajo Fin de Grado han sido definidos por IETF [32] (*Internet Engineering Task Force*), este organismo permite la utilización libre de los documentos y están publicados en Internet.

B. OpenWRT y configuración de Router

B.1. Introducción

OpenWRT [5] es una distribución Linux diseñada expresamente para routers. OpenWRT es un sistema de archivos completamente modificable con gestión de paquetes. La gran ventaja que ofrece OpenWRT respecto a los firmware de fábrica de los routers es la posibilidad de customizar el sistema operativo de los routers, añadiendo o eliminando funcionalidades para realizar una imagen a medida. OpenOWRT también posee un sistema de paquetes que ofrece a los usuarios la posibilidad de desarrollar aplicaciones y añadirlas de forma sencilla sin tener que desarrollar un firmware completo.

OpenWRT no es compatible con todos los routers del mercado, existe un listado en la página web oficial donde se detallan todos los modelos compatibles. Inicialmente en este Trabajo Fin de Grado se utilizó el modelo Linksys WRT54GL, que por falta de potencia fue finalmente sustituido por el Asus WL-500G Premium con tarjeta de red Atheros.

B.2. Creación de imagen a medida

La instalación de OpenWRT en el router es muy sencilla, primeramente es necesario tener una imagen de OpenWRT compatible con el router. Esta imagen puede ser una imagen genérica, que se puede descargar de la página web, o una imagen hecha a medida mediante la herramientas que ofrece OpenWRT.

En este Trabajo Fin de Grado se ha creado una imagen a medida para el Asus WL-500G Premium con todas las características deseadas, para ello se ha tenido que instalar el buildroot OpenWRT en las máquinas del laboratorio. Existen diferentes versiones de OpenWRT, en este proyecto se ha utilizado la versión más reciente 14.07 Barrier Breaker [7]. Para la instalación del Buildroot [23] de OpenWRT se han seguido los siguientes pasos:

1) Instalación de subversion para descargar código fuente:

   sudo apt-get update

   sudo apt-get install subversion build-essential

2) Descarga de las fuentes de OpenWRT mediante subversion, para ello crearemos un directorio llamado openwrt donde se descargarán todos los ficheros:

   mkdir ~/openwrt

   cd ~/openwrt

svn co svn://svn.openwrt.org/openwrt/trunk/

cd trunk

3) Descarga e instalación de feeds, una vez dentro del directorio ~/openwrt/trunk ejecutaremos:

   ./scripts/feeds update -a

   ./scripts/feeds install -a

4) Finalmente se comprobará que no faltan paquetes con los siguientes comandos:

   make defconfig

   make prereq

   make menuconfig

5) En caso de faltar algún paquete se instalará mediante el gestor de paquetes. A continuación se muestra una línea de ejemplo para un equipo Debian en la que se instalan los paquetes gawk, ncurses-dev unzip y zlib1g-dev:

   aptitude install gawk ncurses-dev unzip zlib1g-dev

Una vez instalado el buildroot de OpenWRT podemos crear una imagen a medida con los paquetes y características que queramos. Para crear estas imágenes se utiliza el Buildroot [24] de OpenWRT, una herramienta que mediante un sencillo menú nos permitirá elegir los módulos del kernel, librerías y funcionalidades que queremos introducir en nuestra imagen, cuales no queremos introducir y cuales queremos de manera modular en forma de paquetes.

Para entrar en el Buildroot y crear la imagen a medida es necesario utilizar estos comandos:

cd ~/openwrt/trunk

make menuconfig

A continuación se muestra un menú de ejemplo del Buildroot de OpenWRT:

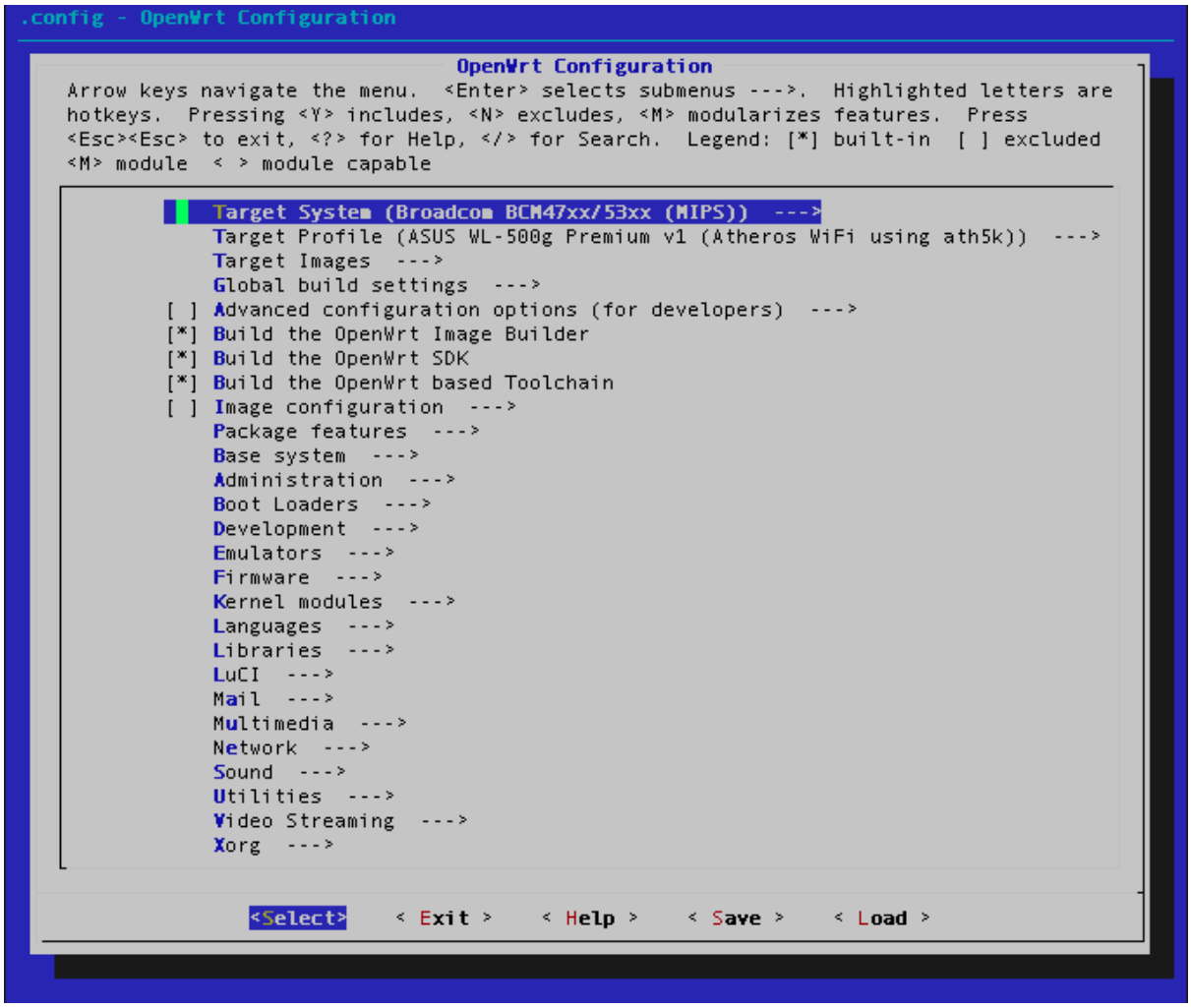


Ilustración 55: Menú de *buildroot* de OpenWRT

En este Trabajo Fin de Grado se ha creado una imagen con todas las librerías y funcionalidades IPv6 necesarias para el funcionamiento de PMIPv6 y mcproxy y se han eliminado todos los paquetes relacionados con el *firewall* para hacer la imagen más liviana. También se han utilizado todas las funcionalidades correspondientes al protocolo 802.11 en forma de paquetes ya que no todos los routers utilizados necesitan de conexión inalámbrica (LMA).

Una vez elegida y guardada la configuración del Buildroot se creará la imagen deseada con el siguiente comando:

make V=s -j3

El ordenador procederá a compilar todo lo necesario para crear la imagen a medida, al terminar terminar el proceso se habrá creado diferentes archivos correspondientes a la imagen y los paquetes. En nuestro caso concreto la imagen se genera en:

~/openwrt/trunk/bin/brcm47xx/

Y los paquetes en:

~/openwrt/trunk/bin/brcm47xx/packages

B.3. Instalación de OpenWRT en el router

La instalación de la distribución en los routers se puede hacer de varias maneras, y con diferentes sistemas operativos, en este Trabajo Fin de Grado se han utilizado dos métodos diferentes desde una máquina Linux. Antes de comenzar la instalación del firmware OpenWRT hay que realizar una serie de preparativos:

- Tener conectado el router al ordenador mediante cable ethernet.
- Tener configurada una IP del rango por defecto del router (192.168.1.1/24).
- Se debe de arrancar el router en modo Failsafe, para ello desconecte el cable de energía del router y conecte mediante un cable ethernet el router al ordenador. Conecte de nuevo el cable de energía y espere a que el LED llamado POWER se apague, cuando se apague pulse el botón *reset* situado en la parte trasera del router varias veces. El LED llamado POWER comenzará a parpadear, esto indica que el router está en modo *Failsafe*. El router ahora debería tener conexión telnet a la IP 192.168.1.1.

Una vez estén preparados el router y la máquina Linux para la instalación de OpenWRT podemos utilizar dos métodos. A continuación se muestran los dos métodos utilizados con sus respectivos comandos, estos comandos pueden contener direcciones IP, nombres de usuarios, directorios o nombres de archivos utilizados en este Trabajo Fin de Grado en concreto y puede que difieran a otros ejemplos.

1) Método TFTP: esta instalación se realiza a través de una conexión TFTP [33] con el router. Se introducirán los comandos mostrados a continuación en la terminal, una vez introducidos hay que desconectar y conectar el cable de alimentación para que comience la transferencia del archivo:

   tftp 192.168.1.1

   tftp> binary

   tftp> rexmt 1

   tftp> timeout 60

   tftp> trace

   tftp> put openwrt-brcm47xx-squashfs.trx

2) El siguiente método es mediante MTD: en este método nos conectaremos remotamente al router, copiaremos la imagen a la memoria del router y ejecutaremos el comando MTD [34] de instalación.

telnet 192.168.1.1

mount_root

scp -p sgonzalez@192.168.1.2:Descargas/*.trx /tmp/

cd /tmp/

mtd write openwrt-brcm47xx-squashfs.trx linux & reboot

B.4. Instalación y compilación de paquetes

Una de las grandes ventajas de OpenWRT es la instalación de software de manera modular mediante paquetes. Los paquetes pueden ser compilados en el buildroot junto a la imagen, o descargados mediante alguna web o sistema subversion. Para compilar una funcionalidad como paquete sólo hay que marcarla con M en el buildroot y compilarlo junto a la imagen con el comando:

make V=s -j3

También es posible compilar el paquete deseado de manera independiente [35] con el comando:

make package/nombredelpaquete/compile V=s

Una vez compilados los paquetes se encontrarán, en nuestro caso, en el siguiente directorio:

~/openwrt/trunk/bin/brcm47xx/packages

Ahora podemos proceder a su instalación [36] en el router OpenWRT. Para ello primeramente copiaremos remotamente el paquete compilado y una vez dentro del router ejecutaremos el comando de instalación:

telnet 192.168.1.1

scp -p sgonzalez@192.168.1.2:Descargas/mcproxy*.ipk /tmp/

cd /tmp/

opkg install mcproxy*.ipk

También pueden instalarse paquetes definidos por OpenWRT o paquetes de una url con el siguiente comando (aunque este método puede lanzar errores de incompatibilidad):

opkg install mcproxy

opkg install http://downloads.openwrt.org/barrier_breaker/14.07/brcm47xx/generic/packages/routing/mcproxy_2014-05-02-bbb2e7ee230c172e68766946e4b4e48f7449ee15-1_brcm47xx.ipk

B.5. Compilación cruzada de código para OpenWRT

OpenWRT es una distribución Linux por lo que ofrece la posibilidad de compilar código en cualquier lenguaje para su ejecución en un router. En nuestro trabajo Fin de Grado se ha compilado una implementación del protocolo PMIPv6 desarrollada en el lenguaje de programación C. Para realizar la compilación cruzada [8] del código nos hemos colocado en el directorio base de la implementación de PMIPv6 y se han seguido los siguientes pasos:

1) Añadir binarios y toolchain al PATH de linux:

   PATH=$PATH:/home/sgonzalez/openwrt/trunk/staging_dir/toolchain-mipsel_mips32_gcc-4.8-linaro_uClibc-0.9.33.2

2) Ajustar las variable de entorno CFLAGS y LDFLAGS:

   CFLAGS='-I /home/sgonzalez/openwrt/trunk/staging_dir/target-mipsel_mips32_uClibc-0.9.33.2/usr/include/'

   LDFLAGS='-L /home/sgonzalez/openwrt/trunk/staging_dir/target-mipsel_mips32_uClibc-0.9.33.2/usr/lib/'

3) Ejecutar configure:

   ./configure --enable-vt --host=mipsel-openwrt-linux-uclibc CFLAGS='-I /home/sgonzalez/openwrt/trunk/staging_dir/target-mipsel_mips32_uClibc-0.9.33.2/usr/include/' LDFLAGS='-L /home/sgonzalez/openwrt/trunk/staging_dir/target-mipsel_mips32_uClibc-0.9.33.2/usr/lib/'

4) Ejecutar make:

   make CC=mipsel-openwrt-linux-uclibc-gcc LD=mipsel-openwrt-linux-uclibc-ld

Una vez ejecutados todos estos pasos se creará un archivo llamado pmip6d en el directorio origen de la implemetación PMIPv6. Este ejecutable está compilado y listo para ejecutar en un router con entorno OpenWRT. Procederemos a copiarlo al router para probar su ejecución y testear si es necesaria la instalación de algún tipo de librería.

B.6. Configuración del Router OpenWRT

Una vez instalado el software correspondiente en nuestros Asus WL-500G Premium podremos proceder con su configuración. En el entorno OpenWRT se puede realizar la configuración de dos maneras diferentes. La primera y más costosa es mediante la introducción de comandos en consola, este tipo de configuración volátil y a la vez poco productiva ya que cada vez que se apague o reinicie el router se perderá toda la configuración. El otro método, el utilizado en este Trabajo Fin de Grado, se basa en la edición de los archivos de configuración del router, este método a diferencia del anterior conservará la configuración después del reinicio y además podrá ser guardada como *backup* copiando este archivo para su instalación en otros routers.

A continuación vamos a proceder a la explicación de la configuración utilizada bajo el segundo método. Existen dos configuraciones, la configuración de red y la configuración inalámbrica. Esta segunda no será necesaria en el router que funcione como LMA ya que no necesitará este tipo de conexión. La configuración de red [37] está localizada en el archivo **etc/config/network** aquí se pueden configurar las interfaces (direcciones IP, rutas, etc) este archivo será modificado en los tres routers de nuestro escenario(MAG1, MAG2 y LMA). La configuración de la conectividad inalámbrica está localizada en el archivo **etc/config/wireless**, este archivo será modificado únicamente en los dos MAG. A continuación vamos a mostrar una imagen de las conexiones traseras del router junto a su arquitectura interna para entender mejor la configuración:

Ilustración 56: Parte trasera del Asus WL-500G Premium V1

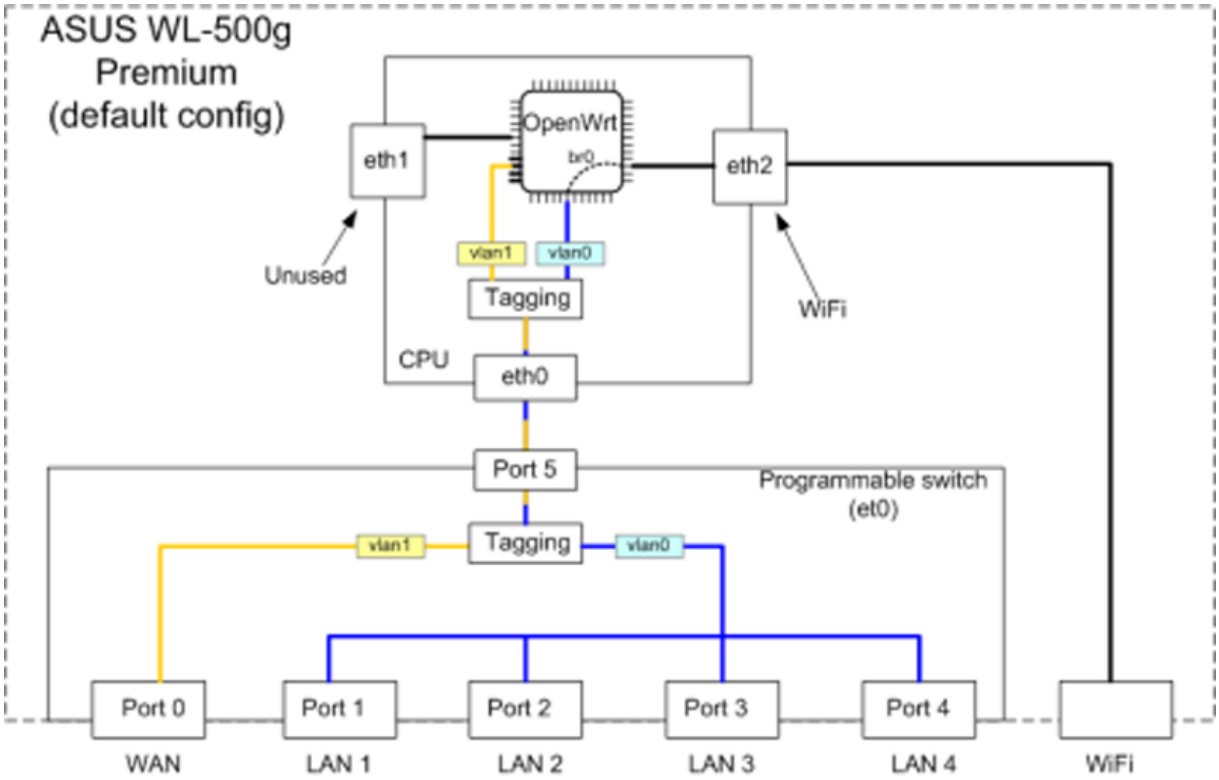


Ilustración 57: Arquitectura interna de Asus WL-500G Premium V1

Procederemos a configurar de esta manera los tres routers de nuestro escenario:

- eth0.0 en el puerto WAN para conexión a Internet.
- En los MAG los puertos LAN 2, LAN3 y LAN4 están unidos como eth0.1. Se configurarán dos direcciones de manera estática: una IPv4 y una IPv6. La IPv4 será utilizada para interconectar todos los routers y así poder configurarlos desde el PC. La dirección IPv6 será la que utilizaremos para la comunicación a través del protocolo PMIPv6. Estas direcciones variarán de un router a otro.
- En el LMA se configurarán dos interfaces diferentes una en LAN 2 y LAN 3 llamada eth0.1 con funcionamiento igual al mencionado en el anterior párrafo y se diferenciará otra interfaz eth0.4 que utilizará el puerto LAN 4, esta interfaz sólo tendrá configurada una IPv4 para configuración y por ella transmitirá los datos el CN.
- eth0.2 en el puerto LAN 1. Se configurará en todos los routers la misma IP estática por defecto usada para configuración: 192.168.1.1/24.
- Sólo se configurará interfaz inalámbrica en los routers MAG1 y MAG2 ya que el router LMA no hace uso de esta interfaz.

Ya que la configuración de red e inálambrica de MAG1 y MAG2 son prácticamente iguales exceptuando las direcciones IP asignadas se va mostrar una como ejemplo. También se va a mostrar la configuración de red del LMA ya que difiere un poco a la de los MAG.

A continuación vamos a mostrar los archivos de configuración del MAG1:

etc/config/wireless

config wifi-device wifi0

option type atheros

option channel 3

option hwmode 11g

option macaddr 00:c0:ca:1b:34:29

option ip6addr '2000:1::1/64'

 # REMOVE THIS LINE TO ENABLE WIFI:

option disabled 0

config wifi-iface

option device wifi0

option network wlan

option mode ap

option ssid MAG1

option encryption none

La configuración de la interfaz inalámbrica difiere un poco de las configuraciones “estándar”, esto es debido a que el router tiene cambiada la tarjeta de red original por una tarjeta de red Atheros. Esto es una modificación muy habitual del router Asus WL-500G Premium.

En cuanto a las diferencias entre la configuración inalámbrica de MAG1 y MAG2 cabe destacar que se han utilizado canales distintos, en MAG1 se ha utilizado el canal 3 y en MAG 2 el canal 11, ya que si se utilizan los mismos canales puede haber conflictos entre los routers.

etc/config/network

```
config switch 'eth0'
        option name 'switch0'
        option reset '1'
        option enable_vlan '1'

config switch_vlan 'eth0_0'
        option device 'switch0'
        option vlan '0'
        option ports '0 5t'

config switch_vlan 'eth0_1'
        option device 'switch0'
        option vlan '1'
        option ports '2 3 4 5t'

config switch_vlan 'eth0_2'
        option device 'switch0'
        option vlan '2'
        option ports '1 5t'

config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan1'
        option ifname 'eth0.1'
        option proto 'static'
        option ipaddr '10.0.0.1/24'
        option ip6addr '2000::1/64'

config interface 'lan2'
        option ifname 'eth0.2'
        option proto 'static'
        option ipaddr '192.168.1.1/24'

config interface 'wlan'
        option proto 'static'
        option ipaddr '10.1.0.1/24'
        option ip6addr '2000:1::1/64'

config interface 'wan'
        option ifname 'eth0.0'
        option proto 'dhcp'

config interface 'wan6'
        option ifname '@wan'
        option proto 'dhcpv6'
```

A continuación vamos a mostrar el archivo de configuración de red del LMA:

etc/config/network

```
config switch 'eth0'
        option name 'switch0'
        option reset '1'
        option enable_vlan '1'

config switch_vlan 'eth0_0'
        option device 'switch0'
        option vlan '0'
        option ports '0 5t'

config switch_vlan 'eth0_2'
        option device 'switch0'
        option vlan '2'
        option ports '2 3 5t'

config switch_vlan 'eth0_1'
        option device 'switch0'
        option vlan '1'
        option ports '1 5t'

config switch_vlan 'eth0_4'
    option device 'switch0'
    option vlan '4'
    option ports '4 5t'

config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan2'
        option ifname 'eth0.2'
        option proto 'static'
        option ipaddr '10.0.0.3/24'
        option ip6addr '2000::3/64'

config interface 'lan4'
    option ifname 'eth0.4'
    option proto 'static'
    option ipaddr '10.10.0.3/24'

config interface 'lan1'
        option ifname 'eth0.1'
        option proto 'static'
        option ipaddr '192.168.1.1/24'

config interface 'wan'
        option ifname 'eth0.0'
        option proto 'dhcp'

config interface 'wan6'
        option ifname '@wan'
        option proto 'dhcpv6'
```

B.7. Configuración PMIPv6 y mcproxy

Una vez configurado el router y sus interfaces correctamente procederemos a realizar las configuraciones necesarias para que puedan funcionar correctamente la implementación de PMIPv6 y mcproxy.

B.7.1. Configuración PMIPv6

Es necesario que las dos interfaces inalámbricas (MAG1 y MAG2) tengan configurada la misma dirección MAC por lo que se introducirá el siguiente comando:

sudo ip link set athwifi0 down

sudo ip link set athwifi0 addr <MAC_address>

sudo ip link set athwifi0 up

Para que PMIPv6 funcione de manera correcta en los MAG necesita un archivo de configuración llamado match, se situará en el directorio tmp, este archivo contiene el prefijo de red (HNP) y la dirección MAC de los nodos móviles autorizados en el dominio PMIPv6. En este archivo cada línea representa una entrada correspondiente a un MN, la primera parte de la línea señala el HNP y la segunda la dirección MAC del MN. En nuestro caso introduciremos la dirección MAC de nuestro MN y el prefijo de red que utilizaremos será 2200::/64, quedando el fichero *match* con esta línea:

22000000000000000000000000000000 00000000000000000000095bc86501

Puede que en algún caso el MAG no detecte correctamente la conexión del MN, esto se produce porque no se ha enviado ningún *Router Solicitation*, se puede forzar el envío del RS en el MN con este comando:

sudo rdisc6 athwifi0

B.7.2. Configuración mcproxy

Una vez instalado mcproxy [9] necesita de un archivo de configuración para su ejecución, llamado mcproxy.conf, en este archivo se indicarán las interfaces de entrada de tráfico multicast y las interfaces por las que replicará este tráfico, también se detalla el protocolo con el que se desea que funcione mcproxy (IPv6 y MLDv2 o IPv4 e IGMPv3). El archivo contiene líneas comentadas que muestran otras opciones posibles y ejemplos. A continuación se muestra el contenido del archivo de configuración utilizado en el router LMA:

```
########################################
##-- mcproxy configuration script --##
########################################

protocol MLDv2; #IPv6
#protocol IGMPv3; #IPv4

pinstance myProxy: "eth0.4" ==> "eth0.2";
#pinstance my_second_instance: tun1 ==> "vlan-eth0.2";

#
# This confiugration example creates
# a multicast proxy for ipv4 with the
# upstream eth0 and two downstreams.
#
#               |
#               |
#        +------+-----+
#        |     eth0    |
#        |             |
#        |  myProxy    |
#        |             |
#        | eth1  eth2  |
#        +---+----+---+
#            |    |
#            |    |
#
```

Ilustración 58: Archivo de configuración de mcproxy

C. Instalación y configuración de programas útiles

C.1. Introducción

Para la realización de este Trabajo fin de Grado se han utilizado diferentes programas que han requerido de una instalación previa, en este anexo se explican todos los pasos realizados para la instalación de estos programas.

C.2. SMCRoute

SMCRoute (*Static Multicast Route*) es una herramienta de línea de comandos que vamos a utilizar para el envío de paquetes multicast. Dentro de este demonio vamos a utilizar la herramienta de testeo de tráfico multicast llamada MCSender. Para su instalación completa ejecutaremos el siguiente comando en la terminal de nuestra máquina Linux:

sudo apt-get install smcroute

C.3. SSMPing

SSMPing es una herramienta para el testeo de la recepción de tráfico multicast. En este Trabajo Fin de Grado utilizaremos una herramienta integrada en el paquete de SSMPing, su nombre es MCFirst, y será la encargada de la recepción del tráfico enviado por MCSender. Para la instalación completa se ejecutará el siguiente comando en la terminal de nuestra máquina Linux:

sudo apt-get install ssmping

C.4. Wireshark

Wireshark es una herramienta que permite la captura y análisis en vivo del tráfico de una red. Esta herramienta se ha utilizado en este Trabajo Fin de Grado para comprobar los intercambios de mensajes y la composición de éstos. Para la instalación de esta herramienta se utilizará el siguiente comando en la terminal de la máquina Linux:

sudo apt-get install wireshark

Bibliografía

[1]: CISCO, The Cisco® Visual Networking Index (VNI) Global Mobile Data Traffic Forecast Update is part of the comprehensive Cisco VNI Forecast, an ongoing initiative to track and forecast the impact of visual networking applications on global networks. This paper presents some of Cisco's major global mobile data traffic projections and growth trends., Febrero 2015, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[2]: C. Perkins, Ed., Tellabs, Inc., D. Johnson, , Rice University, J. Arkko, Ericsson, Mobility Support in IPv6. RFC 6275 (Standards Track), Julio 2011, https://tools.ietf.org/html/rfc6275

[3]: S. Gundavelli, Ed., K. Leung, Cisco, V. Devarapalli, Wichorus, K. Chowdhury, Starent Networks, B. Patil, Nokia, Proxy Mobile IPv6. RFC 5213 (Standards Track), Agosto 2008, http://tools.ietf.org/html/rfc5213

[4]: H. Holbrook, Arastra, Inc., B. Cain, Acopia Networks, B. Haberman, JHU APL, Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast. RFC 4604 (Standards Track), Agosto 2006, http://tools.ietf.org/html/rfc4604

[5]: OpenWRT, https://openwrt.org/, Última consulta: 15/05/2015

[6]: OpenWRT: Linksys WRT54G, http://wiki.openwrt.org/toh/linksys/wrt54g, Última consulta: 03/06/2015

[7]: OpenWRT: Barrier Breaker, http://wiki.openwrt.org/doc/barrier.breaker, Última consulta: 15/06/2015

[8]: OpenWRT: Cross Compile, http://wiki.openwrt.org/doc/devel/crosscompile, Última consulta: 01/06/2015

[9]: Thomas C. Schmidt, Sebastian Wölke, Matthias Wählisch, MCProxy, Peer my Proxy - A Performance Study of Peering Extensions for Multicast in Proxy Mobile IP Domains. In: Proc. of 7th IFIP Wireless and Mobile Networking Conference (WMNC 2014), Piscataway, NJ, USA:IEEEPress, Mayo 2014, https://mcproxy.realmv6.org/trac/

[10]: OpenWRT: Asus WL-500G, http://wiki.openwrt.org/toh/asus/wl500gp, Última consulta: 04/06/2015

[11]: H. Asaeda, Keio University, H. Liu, Q. Wu, Huawei, Tuning the Behavior of the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) for Routers in Mobile and Wireless Networks. RFC 6636 (Informational), Mayo 2012, https://tools.ietf.org/html/rfc6636

[12]: Wikipedia: Agotamiento de las direcciones IPv4, http://es.wikipedia.org/wiki/Agotamiento_de_las_direcciones_IPv4, Última consulta: 03/06/2015

[13]: R. Droms, Bucknell University, Dynamic Host Configuration Protocol. RFC 2131 (Standards Track), Marzo 1997, https://www.ietf.org/rfc/rfc2131.txt

[14]: P. Srisuresh, M. Holdrege, Lucent Technologies, IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Informational), Agosto 1999, https://tools.ietf.org/html/rfc2663

[15]: S. Deering, Cisco, R. Hinden, Nokia, Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Standards Track), December 1998, http://tools.ietf.org/html/rfc2460

[16]: Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291, INTERNET PROTOCOL. RFC 791 (INTERNET STANDARD), Septiembre 1981, http://tools.ietf.org/html/rfc791

[17]: S. Deering, Cisco, R. Hinden, Nokia, Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Standards Track), Diciembre 1998, http://tools.ietf.org/html/rfc2460

[18]: S. Thomson, Cisco, T. Narten, IBM, T. Jinmei, Toshiba, IPv6 Stateless Address Autoconfiguration. RFC 4862 (Standards Track), Septiembre 2007, https://tools.ietf.org/html/rfc4862

[19]: Comisión Nacional de los Mercados y la Competencia (CNMC), Mapa de la cobertura 3G en España (2011), Octubre 2012, http://cnmcblog.es/2012/10/25/mapa-de-la-cobertura-3g-en-espana/

[20]: Comisión Nacional de los Mercados y la Competencia (CNMC), Mapas de la cobertura 3G y 4G en España (2013), Octubre 2014, http://cnmcblog.es/2014/10/14/mapas-de-la-cobertura-3g-y-4g-en-espana-2013/

[21]: Xataka Móvil, El 4G ya supera el 60% de la población y el 3G el 97%. Así quedan los mapas de cobertura, Octubre 2014, http://www.xatakamovil.com/conectividad/el-4g-ya-supera-el-60-de-la-poblacion-y-el-3g-el-97-asi-quedan-los-mapas-de-cobertura

[22]: Wikipedia: IPTV en España, https://es.wikipedia.org/wiki/IPTV#.C2.A0Espa.C3.B1a, Última consulta: 01/06/2015

[23]: OpenWRT: Instalación Buildroot, http://wiki.openwrt.org/es/doc/howto/buildroot.exigence, Última consulta: 03/06/2015

[24]: OpenWRT: Uso Buildroot, http://wiki.openwrt.org/es/doc/howto/build, Última consulta: 06/06/2015

[25]: R. Hinden, Nokia, S. Deering, Cisco Systems, IP Version 6 Addressing Architecture. RFC 4291 (Standards Track), Febrero 2006, https://tools.ietf.org/html/rfc4291

[26]: T. Narten, IBM, E. Nordmark, Sun Microsystems, W. Simpson, Daydreamer, H. Soliman, Elevate Technologies, Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Standards Track), Septiembre 2007, https://tools.ietf.org/html/rfc4861

[27]: B. Haberman, Consultant, D. Thaler, Microsoft, Unicast-Prefix-based IPv6 Multicast Addresses. RFC 3306 (Standards Track), Agosto 2002, https://tools.ietf.org/html/rfc3306

[28]: P. Savola, CSC/FUNET, B. Haberman, JHU APL, Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address. RFC 3956 (Standards Track), Noviembre 2004, https://tools.ietf.org/html/rfc3956

[29]: LM. Contreras, Telefonica I+D, CJ. Bernardos, I. Soto, UC3M, Proxy Mobile IPv6 (PMIPv6) Multicast Handover Optimization by the Subscription Information Acquisition through the LMA (SIAL). RFC 7161 (Experimental), Marzo 2014, http://tools.ietf.org/html/rfc7161

[30]: JC. Zuniga, InterDigital Communications, LLC, LM. Contreras, Telefonica I+D, CJ. Bernardos, UC3M, S. Jeon, Instituto de Telecomunicacoes, Y. Kim, Soongsil University, Multicast Mobility Routing Optimizations for Proxy Mobile IPv6. RFC 7028 (Experimental), Septiembre 2013, https://tools.ietf.org/html/rfc7028

[31]: ABC Tecnología, Nueve de cada diez ingenieros de telecomunicaciones tienen empleo, Febrero 2013, http://www.abc.es/tecnologia/noticias/20130219/abci-trabajo-ofertas-telecomunicaciones-201302181108.html, Última consulta: 22/06/2015

[32]: The Internet Engineering Task Force (IETF®), https://www.ietf.org/, Última consulta: 15/06/2015

[33]: OpenWRT: TFTP, http://wiki.openwrt.org/doc/howto/generic.flashing.tftp, Última consulta: 05/06/2015

[34]: OpenWRT: MTD, http://wiki.openwrt.org/doc/techref/mtd, Última consulta: 02/06/2015

[35]: OpenWRT: Compile Package, http://wiki.openwrt.org/doc/devel/packages, Última consulta: 01/06/2015

[36]: OpenWRT: Where to get packages, http://wiki.openwrt.org/doc/packages, Última consulta: 26/05/2015

[37]: OpenWRT: Network configuration, http://wiki.openwrt.org/doc/uci/network, Última consulta: 01/06/2015